ADL

Homework 3

B07502166 魏子翔

Q1: Model.

1. Model


```
{
  "_name_or_path": "google/mt5-small",
  "architectures": [
    "MT5ForConditionalGeneration"
  ],
  "d_ff": 1024,
  "d_kv": 64,
  "d_model": 512,
  "decoder_start_token_id": 0,
  "dense_act_fn": "gelu_new",
  "dropout_rate": 0.1,
  "eos_token_id": 1,
  "feed_forward_proj": "gated-gelu",
  "initializer_factor": 1.0,
  "is_encoder_decoder": true,
  "is_gated_act": true,
  "layer_norm_epsilon": 1e-06,
  "model_type": "mt5",
  "num_decoder_layers": 8,
  "num_heads": 6,
  "num_layers": 8,
  "pad_token_id": 0,
  "relative_attention_max_distance": 128,
  "relative_attention_num_buckets": 32,
  "tie_word_embeddings": false,
  "tokenizer_class": "T5Tokenizer",
  "torch_dtype": "float32",
  "transformers_version": "4.24.0",
  "use_cache": true,
  "vocab_size": 250112
}
```


MT5 是 T5 模型的多語言版，使用來自MC4之100多種語言數據集訓練。而T5模型與Transformer相似，皆由encoder與decoder構成。在此次的訓練情境下，input爲文章內文(maintext)，output爲每個subword出現之機率。

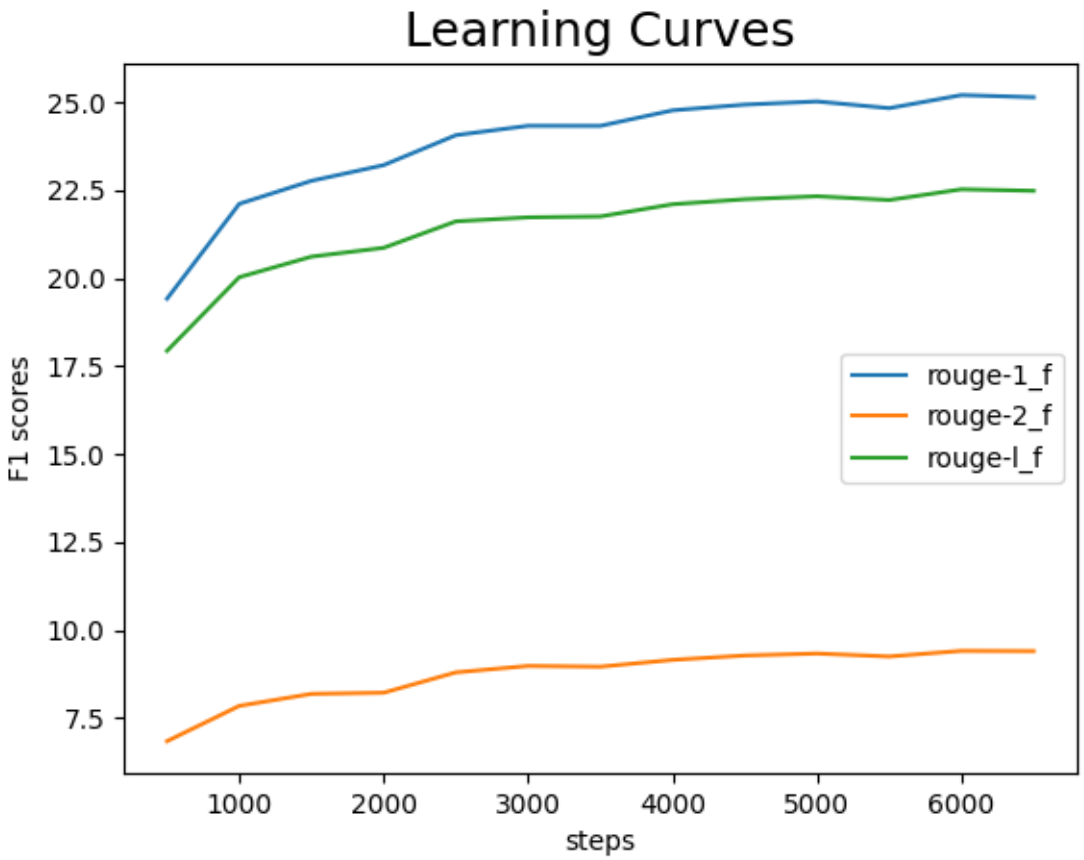
2. Preprocessing

MT5使用的Tokenizer是sentencepiece，其使用byte-pair-encoding 與unigram language model，可將sentence切割成最高機率的subword，並做適當的後處理。

Q2: Training.

1. Hyperparameter
   per_device_train_batch_size=4, gradient_accumulation_steps=4(根據VRAM大小)
   lr=3e-5(根據測試)
   num_epoch=10(根據測試及時間考量)

2. Learning Curves



Q3: Generation Strategies.

1. Strategies
   (1) Greedy: 直接選擇機率最高者
   (2) Beam Search: 每一步驟皆保持N個機率最高者作爲候選，最後再選擇其中最高者
   (3) Top-k: 每一步驟皆從機率分佈前K大者隨機sample出其一
   (4) Top-p: 每一步驟皆從機率分佈由大到小累積至P者隨機sample出其一
   (5) Temperature: 在softmax中加入temperature t以改變vocabulary probability distribution。當t$\to$ 0時爲greedy decoding；當t$\to \infty$時，則爲uniform sampling

2. Hyperparameters

(1) Greedy

strategy	rouge-1_f	rouge-2_f	rouge-L_f
greedy	25.093	9.3624	22.439
sample	20.2158	6.2428	17.6276

從表中可以看出，greedy的結果比隨機sample要好很多，推測是因爲隨機sample並無任何限制，使其經常選到較差之結果，進而影響分數。將此greedy分數與其他strategy做比較亦可發現greedy並不爲一個太差的結果。

(2) Beam Search

strategy	rouge-1_f	rouge-2_f	rouge-L_f
2	26.4221	10.3462	23.5885
3	26.6921	10.5945	23.7148
4	26.6354	10.6663	23.6884
5	26.533	10.6083	23.5828
6	26.5281	10.6813	23.6157

相較於greedy，beam search避免了在計算過程中選錯一步，產生連帶效應後導致的表現下滑，因此其整體表現皆比greedy好了一截。而從表中可以看出，num_beams=3時會有最好的表現，

(3) Top-k

strategy	rouge-1_f	rouge-2_f	rouge-L_f
10	22.6592	7.4765	19.8508
30	20.6602	6.3977	17.9715
50	20.2158	6.2428	17.6276

在k值足夠小的情況下，使用top-k之結果優於隨機sample，但隨著k增大，其效果也越差。推測是因爲k過大之情況下，一次考慮的選項過多，機率較低之選項亦在考慮範圍內，進而sample出表現較差之結果。

(4) Top-p

strategy	rouge-1_f	rouge-2_f	rouge-L_f
0.2	25.0526	9.3756	22.3651
0.5	23.9653	8.5203	21.1648
0.8	22.3392	7.5344	19.5487

使用top-p之結果皆優於隨機sample，但隨著p值增大，表現有下降之趨勢，推測其原因與top-k相似，在p過大的情況下會考慮進機率較低的選項，造成表現下降。

(5) Temperature

strategy	rouge-1_f	rouge-2_f	rouge-L_f
0.2	25.0859	9.3646	22.4321
0.5	25.0783	9.3871	22.4302
0.8	25.1146	9.3773	22.4188

在固定top-p=0.2之條件下，使用不同的temperature皆會使表現略微上升，雖其變動幅度不大，但還是可以顯現出temperature對於模型表現之正面影響。

Final generation strategy: beam search (n_beams=3)