ADL hw2

b10902044

November 8, 2023

Q1: Model

Model

1. architecture: "MT5ForConditionalGeneration"


```
{
  "_name_or_path": "google/mt5-small",
  "architectures": [
    "MT5ForConditionalGeneration"
  ],
  "classifier_dropout": 0.0,
  "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": "4.34.0",
  "use_cache": true,
  "vocab_size": 250112
}
```

The summarization task use the mt5 model, which is similar to conventional vanilla encoder-decoder transformer architecture. Inside the encoder, there are several identical layers, each of which comprises two sub-layers, one for multi-head self-attention and another for a feed-forward network that takes into account the positions of the words in the text.

2. work on text summarization
   The encoder initially maps the source sequences to a sequence of continuous representations. In each decoder step, the given continuous representation sequence and the generated token sequence , the decoder provides a continuous representation to LM head to calculate the logistic regression of each token in the token set. Last, generation strategies are used to find the next token.

Preprocessing

In mt5, the SentencePiece is used to the tokenize the maintext and title. After truncating and padding(all [PAD] in target sequences will be ignored) the source and target sequences to fit the configured maximum lengths, the tokenizer divides sentences into subword units based on dictionary.

Q2: Training

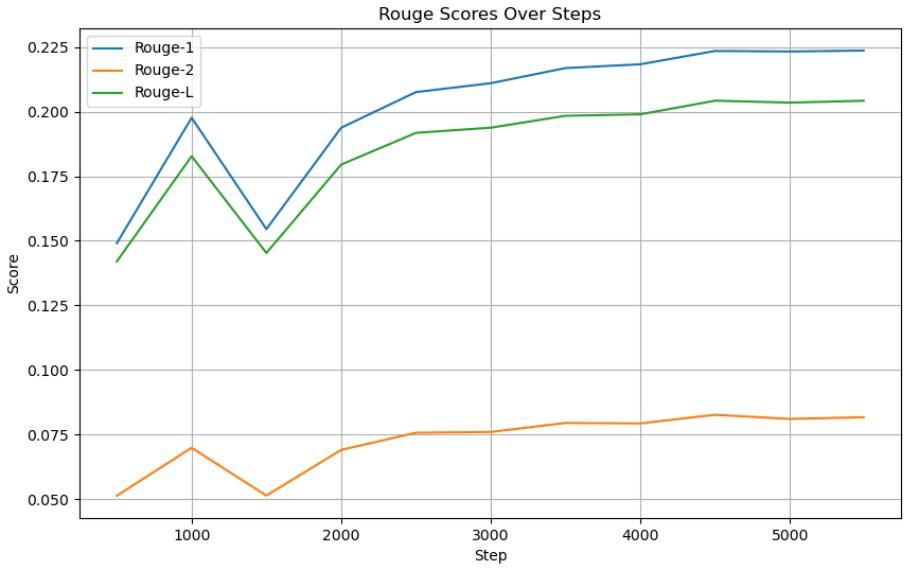
Hyperparameter

1. training epoch: 5
   train the model with enough epoch

2. max_target_length: 128
   the summarization of maintext should be around 128 words(64 is too less and 256 is costly for training)

3. num_beams: 5
   by keeping predicting and evaluating their rouge score

Learning Curves



Q3: Generation Strategies

Strategies

1. Greedy:
   select the word with the highest probability at each step, equal to Beam search with 1

2. Beam Search:
   keep k candidates with the highest probability at each time step and chooses the sequence that has the overall highest probabilities in the end. By selecting the top k words greedily, allows beam search to reduce the risk of missing higher probability word sequences with low probability starting words.

3. Top-k Sampling:
   select one word at each step, but samples it from the top-k words in the output distribution instead of choosing the one with highest probability.

4. Top-p Sampling:
   sample the word from the top-m words which have a probability sum exceeding p.

5. Temperature the temperature divides the numerator and the denominator during the softmax procedure, smaller temperature makes the distribution "harder" and larger temperature makes the distribution "softer", this can effect the result of top-k and top-p sampling strategies.

Hyperparameters

Strategy	Parameters	rouge-1(f)	rouge-2(f)	rouge-l(f)
Greedy	N/A	0.19370466125364114	0.06247289819006101	0.17207395118253674
Beam	beams = 3	0.25584835652388616	0.1001703459962278	0.22875681428397512
Beam	beams = 5	0.2592604923917239	0.10300132003872513	0.23109191233326332
Top-k	k = 10	0.22095776999405983	0.07416026894440787	0.19513598928860382
Top-k	k = 50	0.1960335108096159	0.06267096064315653	0.1729032290505177
Top-p	p = 0.9	0.21020794107475557	0.07072018874679707	0.1856842346280544
Top-p	p = 0.5	0.23705558027367343	0.08615303953674111	0.21155804426167454
Top-p(0.9) + Temperature	temp = 0.7	0.23222993191003158	0.08379470423219267	0.20673876714625739
Top-p(0.9) + Temperature	temp = 2.5	0.09203409977757894	0.012245093201743898	0.0790188972310108

Due to the feature of the text summarization, the strategies involved with sampling don't preform well because if the sampling get bad choice, it can have negative effect to the latter sampling.
Top-k can have better performance than normal sample but if the k value too large, the result will similar to sample.
Top-p also have better performance than normal sample because it have more candidates to choose but if the p value too small, the result will similar to greedy.
Temperature with top-p can have some effect to top-p, if temp < 1, it will sharpen the probability, making less candidate, if temp > 1, it smooth the probability, making top-p more like normal sampling.

In the contrast the beam search keep the candidates and evaluate them in the end, leading to a better performance.

final generation strategy

beams = 5