ADL HW1

b10915024 王程煜

Q1: Data processing

Tokenizer

Describe in detail about the tokenization algorithm you use. You need to explain what it does in your own ways：

首先 tokenizer 會依據我們提供的 max_seq_length 和 padding 來決定是否裁切至某個長度或填充文本，接著對文本進行分割、標記([CLS],[SEP])，最後映射至一個唯一的整數 ID。

Answer Span

How did you convert the answer span start/end position on characters to position on tokens after BERT tokenization?

Tokenizer 會生成 offset mapping，拿取並遍歷 offset mapping，找到答案的 start/end position，並確定對應的起始和結束標記之 index。

After your model predicts the probability of answer span start/end position, what rules did you apply to determine the final start/end position?

遍歷資料集中的所有範例，剔除掉超過 max_seq_length 的資料後，選擇每個位置前 N 個 logits 來識別 N-Best 的起始和結束位置，透過 start 和 end 的機率，計算機率最高的 pair。

Q2: Modeling with BERTs and their variants

Describe

Your model: bert-base-chinese.
The performance of your model: kaggle score 0.76672.
loss function: CrossEntropy loss between the logits and labels.
optimization algorithm: Adam.
learning rate: 3e-5.
batch size: --per_device_train_batch_size 4 --gradient_accumulation_steps 4.

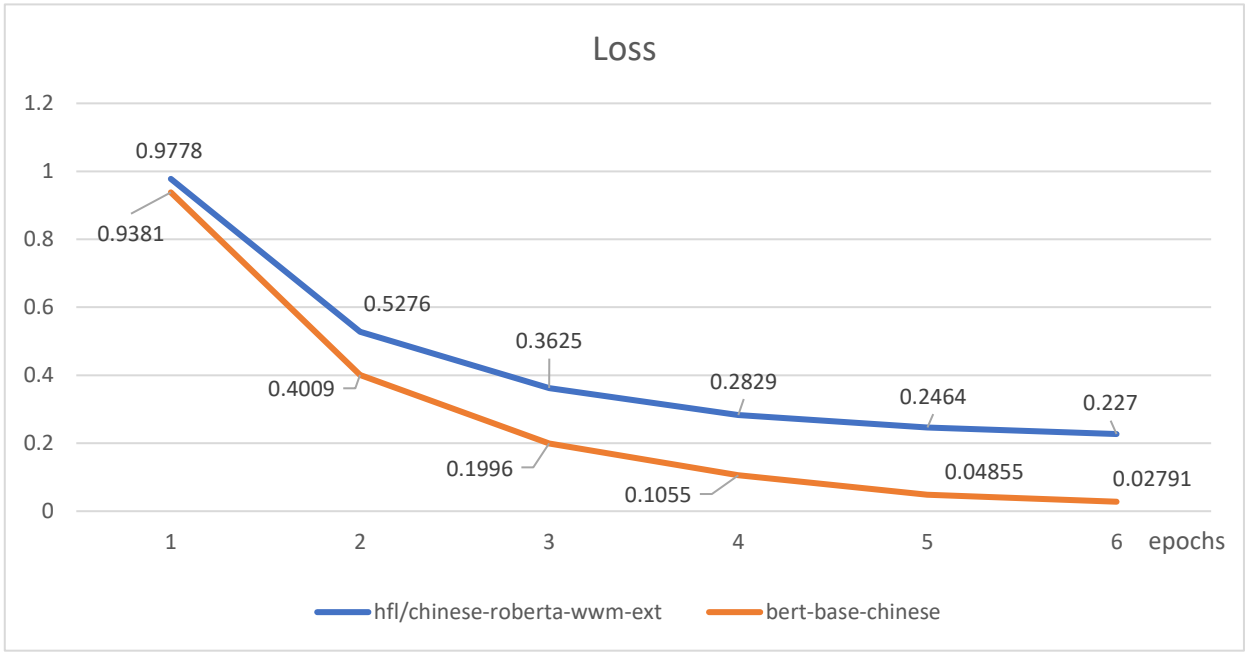
Try another type of pre-trained LMs and describe:

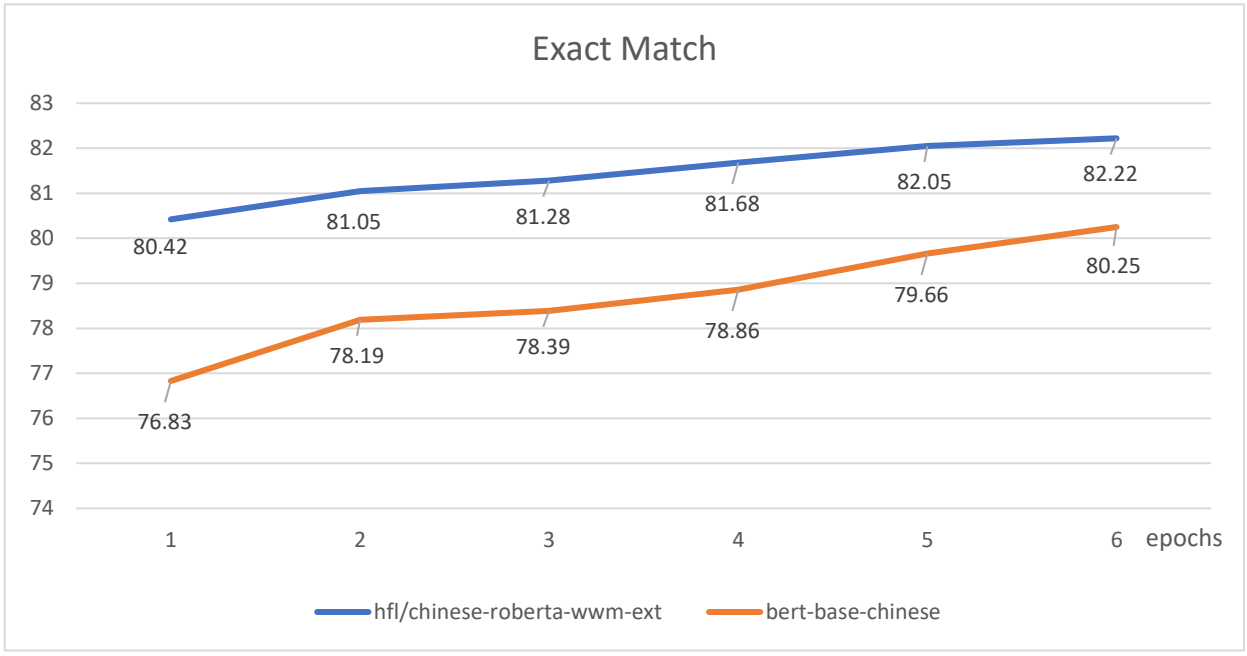
Your model: chinese-roberta-wwm-ext.
The performance of your model: kaggle score 0.79204.
The difference between pre-trained LMs:

Dynamic Masking、Training with large batches、Text Encoding

Q3: Curves

Plot





Q4: Pre-trained vs Not Pre-trained

Describe

The configuration of the model and how do you train this model.


```
{
  "attention_probs_dropout_prob": 0.1,
  "bos_token_id": 0,
  "classifier_dropout": null,
  "directionality": "bidi",
  "eos_token_id": 2,
  "hidden_act": "gelu",
  "hidden_dropout_prob": 0.1,
  "hidden_size": 128,
  "initializer_range": 0.02,
  "intermediate_size": 1500,
  "layer_norm_eps": 1e-12,
  "max_position_embeddings": 512,
  "model_type": "bert",
  "num_attention_heads": 4,
  "num_hidden_layers": 4,
  "output_past": true,
  "pad_token_id": 0,
  "pooler_fc_size": 128,
  "pooler_num_attention_heads": 4,
  "pooler_num_fc_layers": 3,
  "pooler_size_per_head": 32,
  "pooler_type": "first_token_transform",
  "position_embedding_type": "absolute",
  "torch_dtype": "float32",
  "transformers_version": "4.22.2",
  "type_vocab_size": 2,
  "use_cache": true,
  "vocab_size": 21128
}
```


use the qa training command but replace --model_name with --tokenizer_name bert-base-chinese --config_name ./nonpretrain_config.json.

The performance of this model v.s. BERT.
The EM score of bert-base-chinese: 80.25
The EM score of non-pretrain model: 4.32