HW2 chinese QA

R10922132 吳峻銘

Q1. Tokenizer

The tokenization algorithm I use is BERT tokenizer. And, the algorithm behind it is WordPiece. Word Piece first initialize the vocabulary to include every character show in training data and progressively learns a given number of merge rules. This algorithm is similar to BPE. However, there is a little bit different. WordPiece does not choose the most frequent symbol pair, but the one that maximize the likelihood of training data once added to vocabulary.

Q2. Answer span

a. Firstly, I use tokenizer to token the sequence from passage start to the character before start token. And I use these tokenized sequence size plus [cls], [sep], tokenized query size to be start position input. Lastly, I tokenize answer to calculate offset of end position (start+ offset=end).
b. Because, I split the ground truth document into several passage in case that it can't fit model max input size. Hence, a test query may link to several passage. In each case I feed (q, p_i) pair into model. Last, I set following rule to filter unsuitable answer and select the answer with max probability.
    1. Start < end
    2. End -start < 60

Q3. Modeling with bert's and their variants

1. BERT
    A. BertForQuestionAnswering.from_pretrained("bert-base-chinese")
        - Hidden size: 768
        - Number of hidden layers: 12
        - Max input size: 512
        - Activation: gelu

B. Performance
   - Public EM: 0.70343
   - Private EM: 0.71544

C. Loss function
   - Cross entropy

D. Config
   - Optimizer: AdamW
   - Learning rate: 1e-5
   - Weight decay: 1e-4
   - Batch size: 8(per_gpu_training_batch_size) * 2(gradient_accumulation step 2)
   - Epoch: 2

2. MacBERT
   A. AutoModelForQuestionAnswering.from_pretrained("hfl/Chinese-macbert-large")
      - Hidden size: 1024
      - Number of hidden layers: 24
      - Max input size: 512
      - Activation: gelu

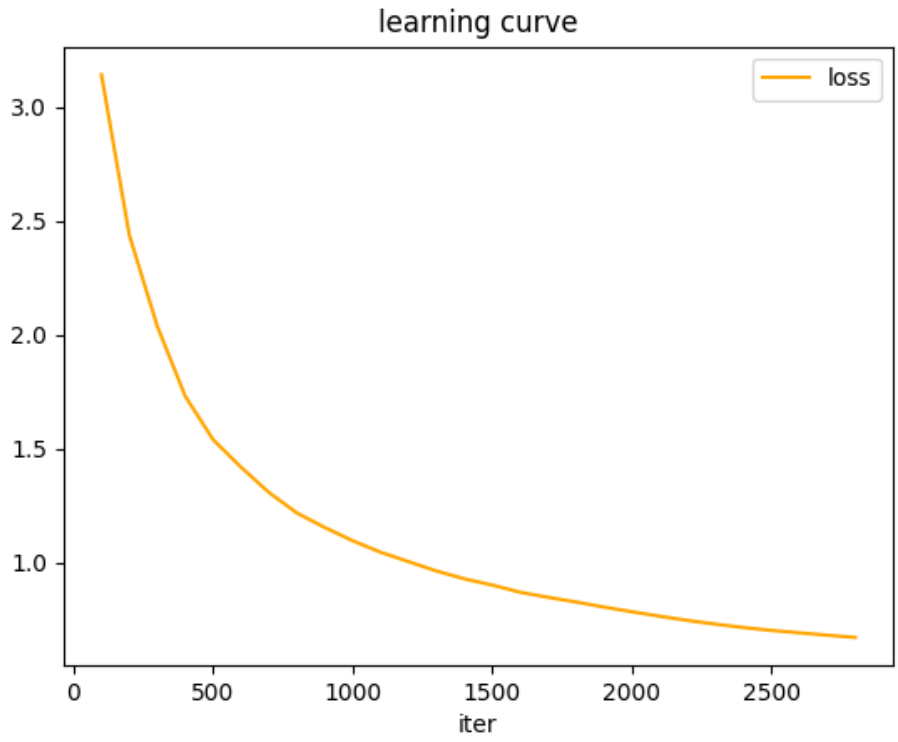
   B. Performance
      - Public EM: 0.77215
      - Private EM: 0.78590

   C. Different
      - The architecture of MacBERT and BERT are the same. The difference is the pre-training task. MacBERT use similar to mask in MLM pretraining task rather than [MASK] token.

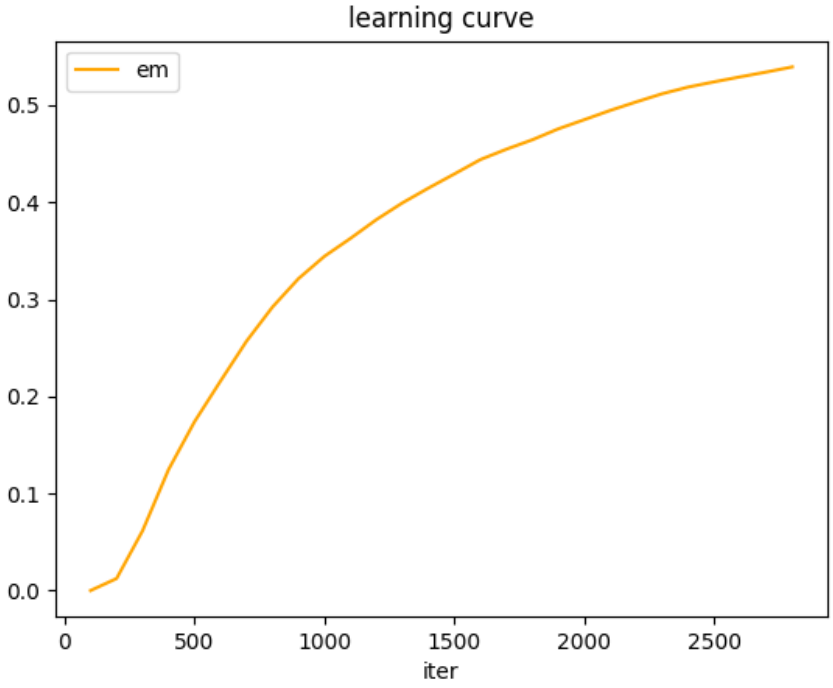
Q4. Curves

The curve is plot in configuration
(bert-base-chinese, epoch 1, batch size 8)

1. Loss



2. Em



Q5. Pretrained vs Not Pretrained

I only load BERT configuration and don't load pretrained weight, so the model configuration is same as BERT

```
if pre_trained_path == None:
    self.config = BertConfig()
    self.bert = AutoModelForQuestionAnswering.from_config(self.config)
```

A. BertForQuestionAnswering (NOT PRETRAINED)
- Hidden size: 768
- Number of hidden layers: 12
- Max input size: 512
- Activation: gelu

B. Performance
- Public EM: 0.02531
- Private EM: 0.02529

C. Loss function
- Cross entropy

D. Config
- Optimizer: AdamW
- Learning rate: 1e-5
- Weight decay: 1e-4

- Batch size: 8(per_gpu_training_batch_size) * 2(gradient_accumulation step 2)
- Epoch: 1

Pretrained is same as Q3-1