Adaptive Deep Learning, 2022 Fall

HW2 Report

R10246002 應數碩二 陳俊憲

Question1：Data processing

1. Tokenizer

- Name: *bert-base-chinese*
- Description: BERT use a algorithm called WordPiece to tokenize texts. Chinese is character-tokenized, which means each character is viewed as a token, as for other language, numbers and signs, WordPiece tokenizer is used.
- WordPiece is a subword segmentation algorithm, it first initialize a vocabulary with all the characters in the text, then use a language model to learn the rule of generate a new word unit by combining two units out of the current vocabulary.

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

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

   Since one example might give us several features if it has a long context, we create an offset_mapping that maps each token to a tuple, which indicates the starting and ending position in the original context, as the example shown below. With this mapping, we can find the relation between the corresponding poistion of the character and the tokenized one. $(0, 0)$ stands for special tokens, like: [CLS]、[SEP]

	Example
context	'《方法論》的作者是誰?'
tokenized	['《', '方', '法', '論', '》', '的', '作', '者', '是', '誰', '?']
offet_mapping	[(0, 0), (0, 1), (1, 2), (2, 3), (3, 4), (4, 5), (5, 6), (6, 7), (7, 8), (8, 9), (9, 10), (10, 11)]

After the probability of starting and ending position of the answer span has been predicted, we compute the following score to determine the top 20 best answer spans,

$$\hat{y}_i^s + \hat{y}_j^e, \text{where } 0 < j - i \leq 30$$

$\hat{y}_i^s$ is the probability of the answer span starts at i^{th} position, $\hat{y}_j^e$ is the probability of the answer span ends at j^{th} position.

Question2：Modeling with BERTs and their variants

1. BERTS

(a) Context Selection

- Model Configuration*
  - bert-base-chinese
  - max_seq_length: 512
- Accuracy: 0.96377
- Loss function
  - Cross-Entropy Loss is used.
- Optimization
  - AdamW with weight decay 1e-5

(b) Question Answering

- Model Configuration*
  - bert-base-chinese
  - max_seq_length: 512
- Accuracy: 0.96377
- Loss function
  - Cross-Entropy Loss is used.
- Optimization
  - AdamW with weight decay 1e-5

- CrossEntropyLoss:

$$L_{context} = -\sum_{i=0}^{3} y_i \log(softmax(\hat{y})) \ ,$$

where $y_i \in \{0, 1\}$, $\hat{y} \in \mathbb{R}^4$ is the output of the context selection BERT.

$$L_{qa} = -\frac{1}{2}\sum_{t=1}^{l_n}[y_t^s \log(softmax(\hat{y}^s)) + y_t^e \log(softmax(\hat{y}^e))] \ ,$$

where $y_t^s, y_t^e \in \{0, 1\}$, which stands for the label of the starting and ending position of answer span, respectively. l_n is the sequence length of batch n. $\hat{y}^s$ and $\hat{y}^e$ are the output of the question answering BERT.

2. Variants of BERT

(a) Context Selection

- Model Configuration*
  - hfl/chinese-bert-wwm-ext
  - max_seq_length: 512
- Accuracy: 0.96776
- Loss function
  - Cross-Entropy Loss is used.
- Optimization
  - AdamW with weight decay 1e-5
  - Epoch: 5, LR: 5e-5

(b) Question Answering

- Model Configuration*
  - hfl/chinese-roberta-wwm-ext-large
  - max_seq_length: 512
- EM: 83.582
- Loss function
  - Cross-Entropy Loss is used.
- Optimization
  - AdamW with weight decay 1e-5
  - Epoch: 5, LR: 5e-5

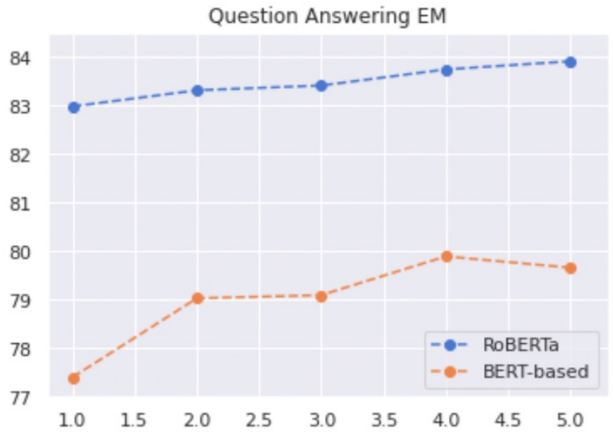
Difference between models

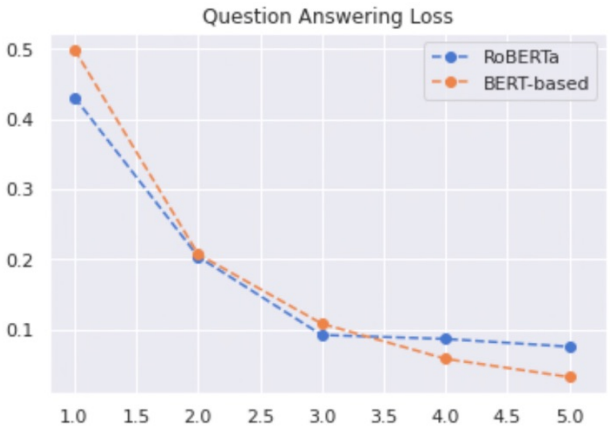
- wwm stands for "whole word masking", unlike the classic masking scheme of BERT, wwm models masks the whole word if one of its subword is masked.
- ext models use more datas for training, training tokens extended to 5.4B.

---

*：For more details of configuration please refer to the last part.

Question3：Curves

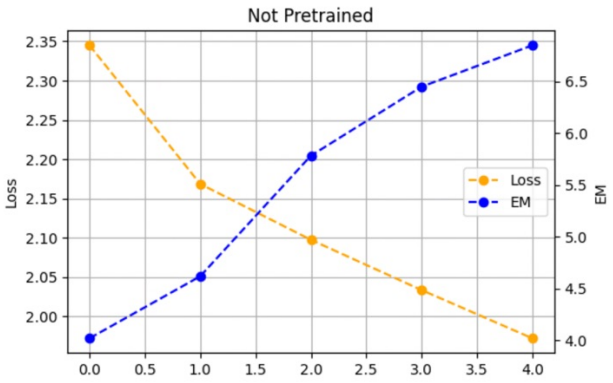


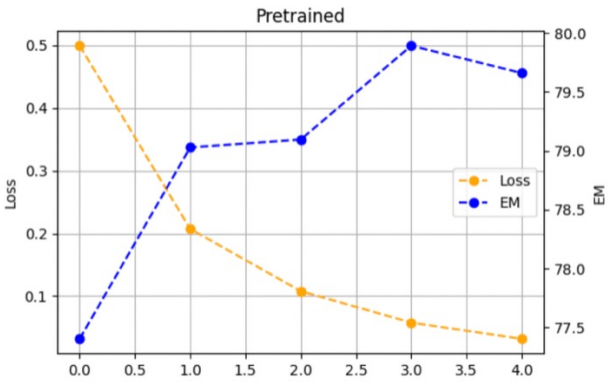


Question4：Pretrained vs Not Pretrained

In this problem, we train a transformer model (bert-based) from scratch on the question answering dataset and compare the performance with that of a pretrained one.

- Model Configuration
  1. Configuration: bert-base-chinese without pretrained model weights
  2. max_seq_length: 512
- Training parameters
  1. Optimizer: AdamW, weight_decay 1e-5
  2. Learning rate: 3e-5, Scheduler: linear
  3. Epoch: 5, Warmup steps: 1
  4. Batch size (Total): 64





Detail Model Configs

```
## bert-base-chinese
{
  "_name_or_path": "bert-base-chinese",
  "architectures": [
    "BertForMaskedLM"
  ],
  "attention_probs_dropout_prob": 0.1,
  "classifier_dropout": null,
  "directionality": "bidi",
  "hidden_act": "gelu",
  "hidden_dropout_prob": 0.1,
  "hidden_size": 768,
  "initializer_range": 0.02,
  "intermediate_size": 3072,
  "layer_norm_eps": 1e-12,
  "max_position_embeddings": 512,
  "model_type": "bert",
  "num_attention_heads": 12,
  "num_hidden_layers": 12,
  "pad_token_id": 0,
  "pooler_fc_size": 768,
  "pooler_num_attention_heads": 12,
  "pooler_num_fc_layers": 3,
  "pooler_size_per_head": 128,
  "pooler_type": "first_token_transform",
  "position_embedding_type": "absolute",
  "transformers_version": "4.22.2",
  "type_vocab_size": 2,
  "use_cache": true,
  "vocab_size": 21128
}
{'_name_or_path': 'hfl/chinese-roberta-wwm-ext-large',
 'architectures': ['BertForQuestionAnswering'],
 'attention_probs_dropout_prob': 0.1,
 'bos_token_id': 0,
 'classifier_dropout': None,
 'directionality': 'bidi',
 'eos_token_id': 2,
 'hidden_act': 'gelu',
 'hidden_dropout_prob': 0.1,
 'hidden_size': 1024,
 'initializer_range': 0.02,
 'intermediate_size': 4096,
 'layer_norm_eps': 1e-12,
 'max_position_embeddings': 512,
 'model_type': 'bert',
 'num_attention_heads': 16,
 'num_hidden_layers': 24,
 'output_past': True,
 'pad_token_id': 0,
 'pooler_fc_size': 768,
 'pooler_num_attention_heads': 12,
 'pooler_num_fc_layers': 3,
 'pooler_size_per_head': 128,
 '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}
```