ADL – HW1

P12922005 林焜詳

Q1 Data Processing

1. Tokenizer

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

在 Bert 中使用的方式為 WordPiece，WordPiece 字面理解是把 word 拆成 piece 一片一片，而課堂上提到的 BPE（Byte-Pair Encoding）即為一種實作的方法，基本步驟如下：

1. 將訓練中所有的單字進行拆解成最小字符單，並建立詞表。
2. 選擇詞表中最相鄰的兩個單詞合併後加入詞表。
3. 重複第二步直到詞表到達需求的量級。

2. Answer Span

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

取得實體化後的 tokenizer，我們可以設定 return_offsets_mapping = True，將答案的開始和結束位置對應到原始上下文。

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?

運用 postprocessing 先去除不可能的答案，去除之後，計算所有機率，並選出有最高機率的語句起始與結束點，成最後的答案。

Q2: Modeling with BERTs and their variants

1. bert-base-chinese (baseline)

a. Configuration

Paragraph Selection	**Question Answering**
{	{
"_name_or_path": "bert-base-chinese",	"_name_or_path": "bert-base-chinese",
"architectures": ["BertForMultipleChoice"],	"architectures": ["BertForQuestionAnswering"],
"attention_probs_dropout_prob": 0.1,	"attention_probs_dropout_prob": 0.1,
"classifier_dropout": null,	"classifier_dropout": null,
"directionality": "bidi",	"directionality": "bidi",
"hidden_act": "gelu",	"hidden_act": "gelu",
"hidden_dropout_prob": 0.1,	"hidden_dropout_prob": 0.1,
"hidden_size": 768,	"hidden_size": 768,
"initializer_range": 0.02,	"initializer_range": 0.02,
"intermediate_size": 3072,	"intermediate_size": 3072,
"layer_norm_eps": 1e-12,	"layer_norm_eps": 1e-12,
"max_position_embeddings": 512,	"max_position_embeddings": 512,
"model_type": "bert",	"model_type": "bert",
"num_attention_heads": 12,	"num_attention_heads": 12,
"num_hidden_layers": 12,	"num_hidden_layers": 12,
"pad_token_id": 0,	"pad_token_id": 0,
"pooler_fc_size": 768,	"pooler_fc_size": 768,
"pooler_num_attention_heads": 12,	"pooler_num_attention_heads": 12,
"pooler_num_fc_layers": 3,	"pooler_num_fc_layers": 3,
"pooler_size_per_head": 128,	"pooler_size_per_head": 128,
"pooler_type": "first_token_transform",	"pooler_type": "first_token_transform",
"position_embedding_type": "absolute",	"position_embedding_type": "absolute",

"torch_dtype": "float32",	"torch_dtype": "float32",
"transformers_version": "4.22.2",	"transformers_version": "4.22.2",
"type_vocab_size": 2,	"type_vocab_size": 2,
"use_cache": true,	"use_cache": true,
"vocab_size": 21128	"vocab_size": 21128
}	}

b. Performance of my model.

Paragraph_Selection_eval_accuray: 0.959122632103689

Question_Ansering_eval_EM: 80.22598870056497

Question_Ansering_eval_F1: 80.22598870056497

Public Result: 0.74683

Private Result: 0.74977

c. Loss function

Cross Entropy Loss

d. The optimization algorithm (e.g., Adam), learning rate and batch size.

Multiple Choice	**Question Answering**
Optimizer: AdamW	Optimizer: AdamW
Learning rate: 3e-5	Learning rate: 3e-5
Batch size: 8	Batch size: 8
weight decay: 0	weight decay: 0
Gradient accumulation: 6	Gradient accumulation: 6

2. Variant Bert (hfl/chinese-macbert-base)

a. Configuration

Paragraph Selection	Question Answering
{	{
"_name_or_path": "hfl/chinese-macbert-base",	"_name_or_path": "hfl/chinese-macbert-base",
"architectures": ["BertForMultipleChoice"],	

```
    "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",
    "torch_dtype": "float32",
    "transformers_version": "4.22.2",
    "type_vocab_size": 2,
    "use_cache": true,
    "vocab_size": 21128
}
```

```
    "architectures":
["BertForQuestionAnswering"],
    "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",
    "torch_dtype": "float32",
    "transformers_version": "4.22.2",
    "type_vocab_size": 2,
    "use_cache": true,
    "vocab_size": 21128
}
```

b. Performance of my model.

Paragraph_Selection_eval_accuray: 0.9687603855101362
Question_Ansering_eval_EM: 81.92090395480226
Question_Ansering_eval_F1: 81.92090395480226
Public Result: 0.80379

Private Result: 0.79674

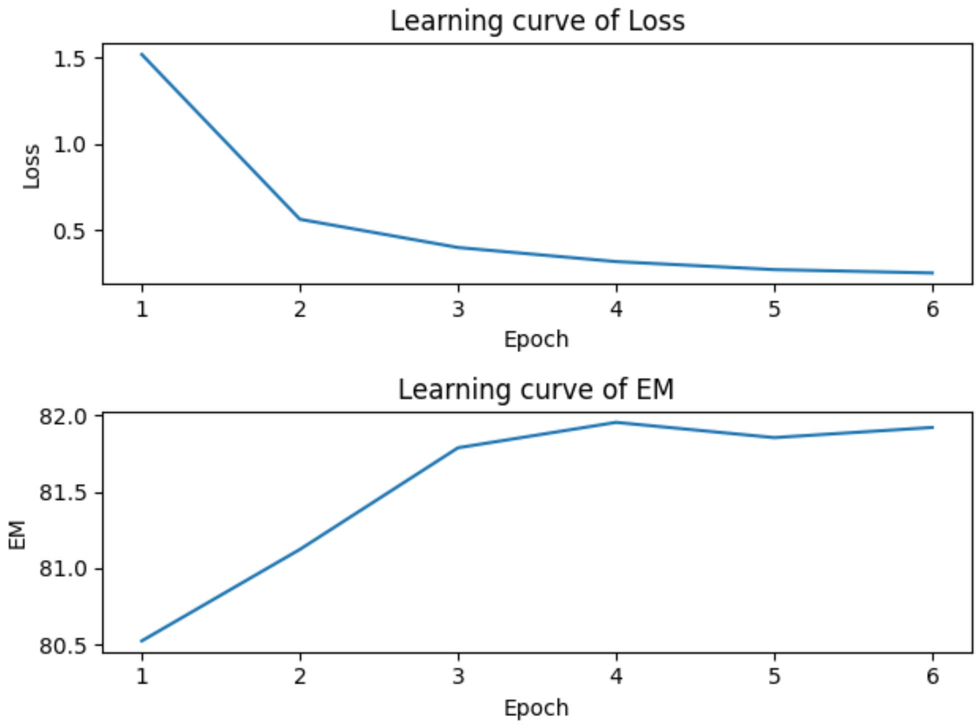
c. The difference between pre-trained LMs (architecture, pretraining loss, etc.)

architecture: MacBert 加入了糾錯型動態掩蓋方法，減少預訓練與下游任務的不一致問題

pretraining loss: MacBert 表現是優於 valina bert 的

Q3. Curve

Model: hfl/chinese-macbert-base



從圖中可知 epoch 4 有最好的表現。另外有發現，一般語言模型任務 fine-tuning 都有使用 weight decay = 0.1 or 0.01，而 hugging face 的預設是 0，且 wd=0 效果有比較好些，這塊還需更多時間去釐清 wd 對這任務的影響在哪。

Q4: Pretrained vs Not Pretrained

此部分我只在 QA 問題上做實驗，使用 bert-base-chinese 模型進行測試，為了要把 Pretrained weight 移除需要把.from_pretrained 修改成 form_config，不讓模型取得預訓練好的資料。

a. Configuration

```
{
  "_name_or_path": "bert-base-chinese",
  "architectures": ["BertForQuestionAnswering"],
  "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",
  "torch_dtype": "float32",
    "transformers_version": "4.22.2",
    "type_vocab_size": 2,
    "use_cache": true,
    "vocab_size": 21128
}
```

b. Performance of model

Paragraph_Selection_eval_accuray: 0.959122632103689

Question_Ansering_eval_EM: 5.81588567630442

Question_Ansering_eval_F1: 5.81588567630442

Public: 0.07775

Private: 0.06684

從上面結果來看，QA model 完全不能用，在 private dataset 上僅有 6.6%的準確度，可能需要從助教建議的方向，將模型變小，或是 train 久一點，同時也凸顯語料模型預訓練的重要性。

Q5: Bonus End to End QA

a. Model

這裡我採用 bert-base-chinese，因需要改成 end to end model，我這邊將每個問題的 paragraph 串連起來，每個 paragraph 用句號連接，因為 context 也跟著變大，所以把 model 需要的 max sequence 的長度方大四倍，512 -> 2048，訓練也跟著變難訓練，目前只有跑兩組實驗，尚未得到如上面好的模型。

b. The performance of my model

Question_Ansering_eval_EM: 32.136922565636425

Question_Ansering_eval_F1: 32.136922565636425

c. The loss function I used

Cross Entropy Loss

d. The optimization algorithm (e.g. Adam), learning rate and batch size.

Optimizer: AdamW

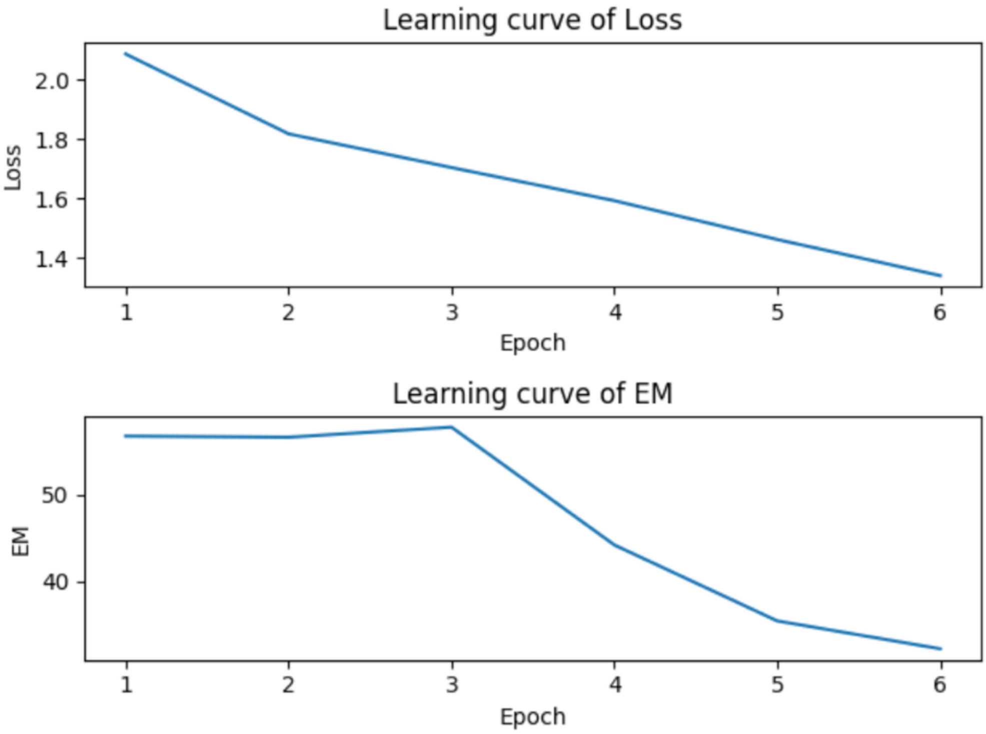
Learning rate: 3e-5

Batch size: 8

weight decay: 0.01

Gradient accumulation: 6

e. 其實表現不好目前訓練起來表現有往下掉的趨勢，部分推測是 vallina bert 在長文本的表現本來就比較差，但因為運算資源，和時間不夠就沒有多做其他實驗了。



PS. Kaggle 上我有試著嘗試使用助教說不能用的模型，發現效果很好，因為有預訓練了，但是上傳後發現不能刪掉，請助教別見怪，我只是想要知道表現會有多好...XD