ADL HW2

首先，我CS和QA的task都是從huggingface的sample script改的，用的都是`no_trainer`的版本。

畫圖與跑code的部分請見readme。

Q1

Tokenizer

> ref.
>
> - https://huggingface.co/course/chapter6/6?fw=pt
> - https://github.com/google-research/bert/blob/master/multilingual.md#tokenization

我們用的tokenizer是huggingface的`BertTokenizer`，其實用的是WordPiece tokenizer。

他的做法跟BPT很像：

- 把所有character (e.g. `['a', 'b', 'c', '##a', '##b', ...]`) 和特殊字元 (e.g. `['[UNK]', '[CLS]', '[SEP]']`) 加到vocabulary。其中如果一個字是以多個token組合而成的話，後面的token會有prefix `##`。
- 開始merge pair。我們算一個分數，並且merge分數最高的那個pair。對於pair `(a, b)`，分數計算為`score = count((a, b)) / (count(a) * count(b))`，概念是盡量merge會常常一起出現而不常獨自出現的token pair。算出分數最大的pair後，把那個pair給merge然後加到vocabulary。e.g. `(##g, ##s)` merge之後變成`##gs`。
- 做出vocabulary之後，對於一個給定的word，我們以vocabulary中最長的subword對一個word去split，直到word被拆成一堆token，或是如果剩下的部分無法被tokenize的話，補`[UNK]`。

不過處理中文等沒有空格的語言的部分，他會在所有CJK Unicode range的字元兩邊加空格，所以在中文上，基本上等同於character tokenized。

Answer Span

> ref.
>
> - https://huggingface.co/course/chapter7/7?fw=tf#postprocessing

在tokenize之後，把answer span的第一個字的第一個token設成start，最後一個字的最後一個token設成end。如果是token要map回去word的話，直接看這個token是屬於哪個word的，就map到那個word。

有probability後，我們在所有pair `(start_logit, end_logit)` 中挑選 `prob(start_logit) * prob(end_logit)` 最大的那個當作答案。不過因為pair數太多，所以他們只挑logit probability最大的幾個操作 (nbest)。

Q2

Base model

CS和QA的model都是一樣的：

```
model='bert-base-chinese'
max_len=512
lr=5e-5
weight_decay=0
num_epoch=1
batch_size=1
grad_acc_step=2
```

- Performance: 0.72322 on kaggle
- Loss function: CrossEntropyLoss (CS, QA)
- Optimization algorithm: AdamW

關於model的更詳細資訊如下：


```
{
  "_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
}
```


Variant model

> ref:
>
> - https://github.com/ymcui/Chinese-BERT-wwm

```
model='hfl/chinese-roberta-wwm-ext'
max_len=512
lr=1e-5 (CS), 2e-5 (QA)
weight_decay=0
num_epoch=1 (CS), 3 (QA)
batch_size=1
grad_acc_step=2
```

- Performance: 0.77576 on kaggle。另外，我是以同樣的setting去train多個模型，看CS和QA組合最好的模型的performance。

差別在於：

- Roberta
  - 有dynamic masking隨著epoch進行改變masking。
  - Optimization方式：調整lr, warmup, batch。
- WWM: whole word masking
  - 把一個word的所有subword一起mask。

WWM的部分，因為上面提到原本的bert對中文等價於以character為token。他們改定義這邊的whole word為中文的「詞」(e.g.「模型」為一詞)，也就是為了masking，他們須先經過分詞才能mask。

關於model的詳細config如下：


```
{
  "_name_or_path": "hfl/chinese-roberta-wwm-ext",
  "architectures": [
    "BertForQuestionAnswering"
  ],
  "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": 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,
  "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",
```

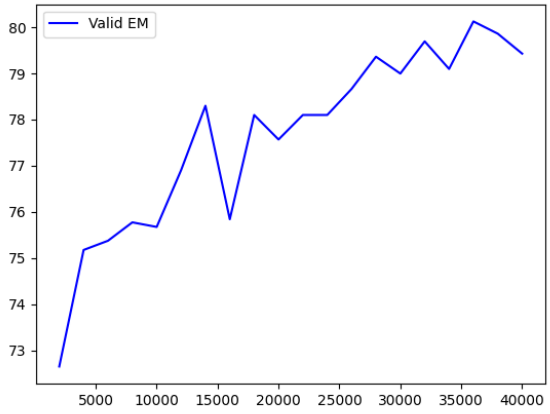
```
  "type_vocab_size": 2,
  "use_cache": true,
  "vocab_size": 21128
}
```

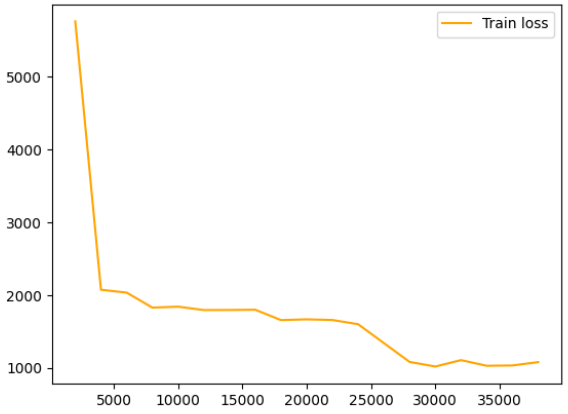
Q3

我每2000 iteration記錄的是validation exact match和train loss (其中一個iteration為更新model一次)。train loss是這2000 iteration的加總。

因為紀錄上，每一個epoch會重新計算目前加總的loss，所以有幾個資料點沒有湊齊2000，在圖中會移除。

這是variant model的QA的圖。





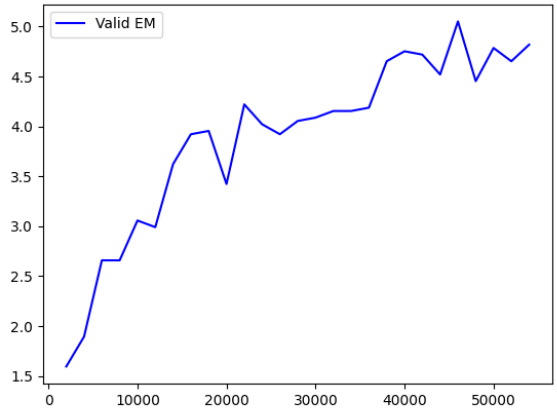
Q4

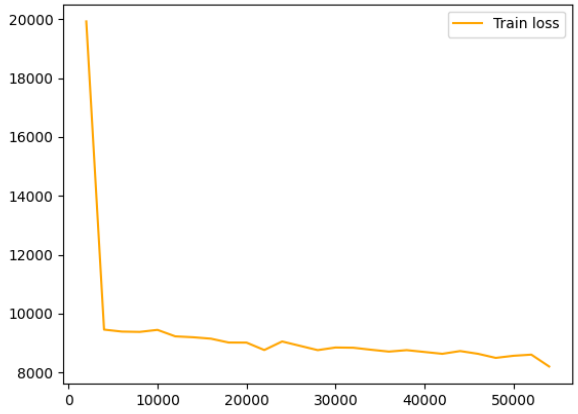
我用的是：

```
model='hfl/chinese-roberta-wwm-ext'
max_len=512
lr=3e-5
weight_decay=0
num_epoch=3
batch_size=1
grad_acc_step=2
```

但是我是直接train上去，roberta-wwm基本上只取架構，關於roberta的訓練方式和WWM都沒有用，所以是用跟fine-tuning一樣的方式train。

這是他的圖：





很有趣的是他跟剛才已經pretrain好的BERT的fine-tuning圖差不多，但是量級差很多。

performance部分valid exact match在大概5%左右，但是BERT和roberta-wwm都是75%以上，甚至光是前面2000 iteration就可以把exact match衝到70%以上。

2000 iteration的total loss部分大概在10000左右，但是BERT和roberta-wwm可以到小於2000，甚至到大約1000。

Q5

N/A