Q1.Data processing

1. Tokenizer

我使用的 model 是 chinese-roberta-wwm-ext-large，其 model_type 為 bert，對應到的 tokenizer 為 Bert Tokenizer，我使用的是 fast 版本，也就是 BertTokenizerFast。在這過程中，它會對 text 執行以下動作。

1.對 text 進行 tokenize，tokenizer 會對 text 使用 wordpiece 進行 subword。Wordpiece 與 BPE 十分相似，最大的差異在於 wordpiece 選擇能夠提升 language model 機率最大的相鄰字詞加入 vocab，BPE 選擇頻率最高的相鄰字詞加入 vocab。

以下是 wordpiece 的 alg.

From: "Japanese and Korean Voice Search "
(Mike Schuster,Kaisuke Nakajima)

1. Initialize the word unit inventory with the base characters.
2. Build a language model on the training data using the word inventory from 1.
3. Generate a new word unit by combining two units out of the current word inventory. The word unit inventory will be incremented by 1 after adding this new word unit. The new word unit is chosen from all the possible ones so that it increases the likelihood of the training data the most when added to the model.
4. Goto 2 until a pre-defined limit of word units is reached or the likelihood increase falls below a certain threshold.

2.在分句之間加入 special tokens(Ex:[CLS]、[SEP])以作區別。

3.對 subword 及 special tokens 進行 convert_tokens_to_ids。

4.如果需要 padding，針對 padding 策略進行 padding 的動作。

5.如果需要 trucation，針對 trucation 策略進行 trucation 的動作。

6.紀錄 convert character to token 及 trucation 造成的位置變化，將轉換資訊存於 offset_mapping 及 sample_mapping 用於轉換位置。sample_mapping 為一個將 text span 對應至原本 example 的 map，因為可能原本 example 內容很長，在處理時必須切割成多個 text span 才能處理，因此需要 sample_mapping 將其對應回去。

offset mapping 為一個能夠將 token 對應至 character position in original context 的 map，我們可以透過此 map 找出所有 token 所對應到的 character。

2. Answer Span

a.

設定 cls_index 用於 impossible answer，start_position 用於紀錄 answer span 在 original text 的起始位置所對應到的第一個 token 的位置，end_position 用於紀錄 answer span 在 original text 的最終位置所對應到的最後一個 token 的位置。

我透過 sample_mapping 找出目前所處理的 text span 所對應的 example，紀錄此 example 的 index 在 sample_index。

找出目前所處理的 text span 所對應的 sequence_ids，sequence_ids 具有辨別 question 及 context 的功能(1 代表 context)。

透過 sample_index 找出此 example 的 answers，透過 answers 中的 start 得到 answer span 在 original text 的起始位置，紀錄在 start_char。紀錄 answer span 在 original text 的最終位置在 end_char。

從 sequence_ids 第一個開始找 sequence_id 為 1 的位置，紀錄 index 在 token_start_index。從 sequence_ids 最後一個開始往前找 sequence_id 為 1 的位置，紀錄 index 在 token_end_index。

Token_start_index 及 token_end_index 代表了 example 中 context 的起始及最終位置所對應到的 token 的起始及最終位置。

有了 token_start_index 及 token_end_index 後，可以先確認 answer 是否位於此 text span 內。如果不在 text span 內，就將 start_position 及 end_position 紀錄為 cls_index。

如果確定了 answer 在此 text span 中，就透過 offsets_mapping 確認 token 所對應到在 context 中的 character。

我們從 token_start_index 開始往後查找，直到找到 answer span 中第一個字所對應到的第一個 token，記錄此 token 位置為 start_position。

我們從 token_end_index 開始往前查找，直到找到 answer span 中最後一個字所對應到的最後一個 token，記錄此 token 位置為 end_position。

我透過以上方法將 answer span 的 start/end position on characters 對應至 start/end position on token。

b.

start_logits 代表每個 token 為 start 的機率，end_logits 代表每個

token 為 end 的機率。

我透過自定義一個 hyperparameter，n_best_size=n，這個 hyperparameter 是用於設定取前 n 個最高機率的 start_logit 及前 n 個最高機率的 end_logit。

我在這之中選出 score 最高的組合作為 answer span 輸出，score 的定義為 start_logit 機率+end_logit 機率。

同時設定條件確認 score 最高的組合是否為 possible answer，如果不是就選次高分數的組合，以此類推。

Q2. Modeling with BERTs and their variants

1. Describe

a.


```
Model config BertConfig {
  "_name_or_path": "./drive/MyDrive/ADL/chinese_bert_MC/",
  "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": "4.24.0",
  "type_vocab_size": 2,
  "use_cache": true,
  "vocab_size": 21128
}
```


上圖為 model 的參數設定，我使用在 Context Selection 的 model 是 bert-base-chinese，在 hyperparameters 我使用了以下設定，其餘皆設定為 default 的參數。

max_length=512、learning_rate=3eˆ-5、weight_decay=0、num_train_epochs=1、lr_scheduler_type=linear、per_device_train_batch_size=1、per_device_eval_batch_size=1、gradient_accumulation_steps=2。

```
Model config BertConfig {
  "_name_or_path": "./drive/MyDrive/ADL/chinese_bert_QA/",
  "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": "4.24.0",
  "type_vocab_size": 2,
  "use_cache": true,
  "vocab_size": 21128
}
```


上圖為 model 的參數設定，我使用在 Question Answering 的 model 是 bert-base-chinese，在 hyperparameters 我使用了以下設定，其餘皆設定為 default 的參數。

max_seq_length=512、learning_rate=3e^-5、weight_decay=0、num_train_epochs=1、lr_scheduler_type=linear、per_device_train_batch_size=1、per_device_eval_batch_size=1、gradient_accumulation_steps=2、n_best_size=20、max_answer_length=60。

b. 在 kaggle 上 public score 為 0.72694，private score 為 0.7308。

c. 兩個 model 的 loss function 都為 CrossEntropy Loss。

d. 兩個 model 的 optimization alg.都是使用 Adam 搭配 linear schedular，learning_rate=3e^-5、batch_size=2。

2. Describe

a.


```
Model config BertConfig {
  "_name_or_path": "./drive/MyDrive/ADL/chinese-roberta-wwm-ext",
  "architectures": [
    "BertForMultipleChoice"
  ],
  "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": "4.24.0",
  "type_vocab_size": 2,
  "use_cache": true,
  "vocab_size": 21128
}
```


上圖為 model 的參數設定，我使用在 Context Selection 的 model 是 chinese-roberta-wwm-ext，在 hyperparameters 我使用了以下設定，其餘皆設定為 default 的參數。

max_length=512、learning_rate=3eˆ-5、weight_decay=0、num_train_epochs=3、lr_scheduler_type=linear、warmup_ratio=0.1、per_device_train_batch_size=1、per_device_eval_batch_size=1、gradient_accumulation_steps=8。


```
Model config BertConfig {
  "_name_or_path": "./drive/MyDrive/ADL/chinese-roberta-wwm-ext-large-qa",
  "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": 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": "4.24.0",
  "type_vocab_size": 2,
  "use_cache": true,
  "vocab_size": 21128
}
```


上圖為 model 的參數設定，我使用在 Question Answering 的 model 是 chinese-roberta-wwm-ext-large，在 hyperparameters 我使用了以下設定，其餘皆設定為 default 的參數。

max_seq_length=512 、 learning_rate=3eˆ-5 、 weight_decay=0 、

num_train_epochs=3 、 lr_scheduler_type=linear 、 per_device_train_batch_size=2 、 per_device_eval_batch_size=2 、 gradient_accumulation_steps=4、n_best_size=20、warmup_ratio=0.1、 max_answer_length=60。

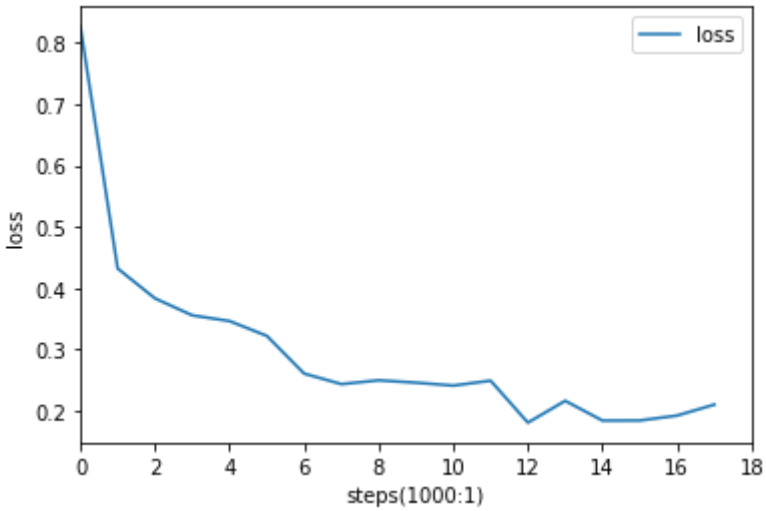
b. 在 kaggle 上 public score 為 0.80018，private score 為 0.80036。兩個 model 的 optimization alg. 都是使用 Adam 搭配 linear schedular，learning_rate=3e^-5、batch_size=8。兩個 model 的 loss function 都為 CrossEntropy Loss。

c. 在不同 pretrained model 之間各有不同的差異，我使用 BERT-base-chinese(以下簡稱 BERT)及 BERT-wwm-ext 作為範例。這兩個模型在架構上是一樣的，但在原始預訓練階段使用不同的 masking 策略，BERT 使用的是對字詞進行 mask，字詞可能不是一個完整有意義的字詞。在 BERT-wwm-ext 中使用的是 Whole Word Masking，對完整的字詞進行 mask。我認為這兩者 mask 上的策略影響了在中文字上的訓練效果，因為中文與英文在字詞結構上的差異使得 BERT 的 mask 策略相較於 BERT-wwm-ext 的 mask 策略造成的學習效果較差，因此造成 BERT-wwm-ext 在中文字相關問題的應用上效果更好。同時 BERT-wwm-ext 也增加了預訓練資料集及訓練 step 數量。

我使用的 RoBERTa 相較於 BERT 做出了更多的改變，例如擴大預訓練數據集、使用 Dynamic Masking、改變 Text Encoding 方式、Training on Longer Sequence 等等。

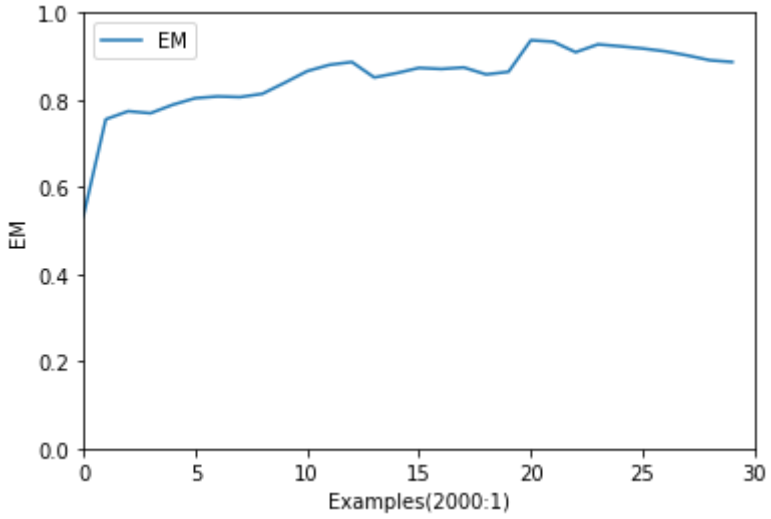
Q3.Curves

a.



上圖為 QA 的 Learning curve of loss，我總共跑了 3 個 epochs，我每跑 1000 個 steps 記錄一次 loss。

b.



上圖為 Learning curve of EM，我記錄了 training 過程中所有的 output，依訓練順序排序，將其與 answer 做比對，每 2000 筆 Example 計算一次 EM。

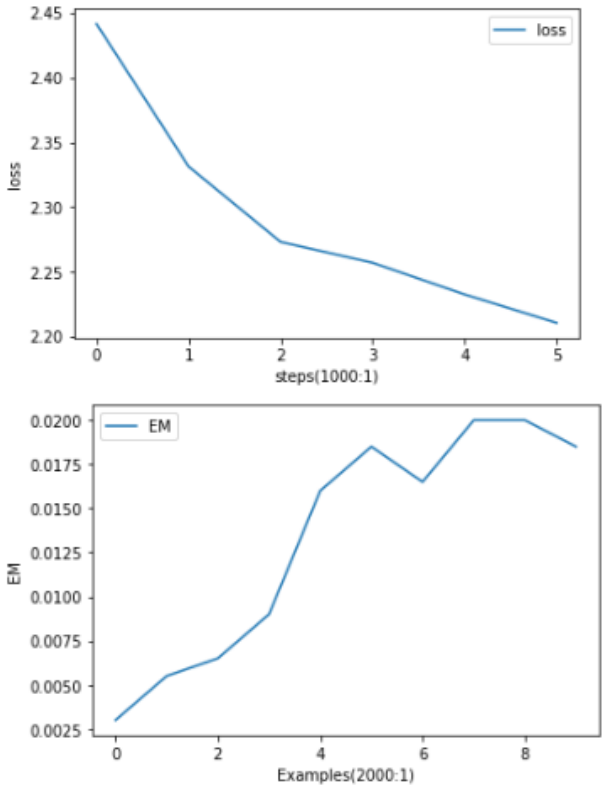
Q4.Pretrained VS not Pretrained


```
Model config BertConfig {
  "_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": "4.24.0",
  "type_vocab_size": 2,
  "use_cache": true,
  "vocab_size": 21128
}
```


我使用 bert-base-chinese 的 config 建構 model，我選擇的是 QA。上圖為 model 的參數設定，在 hyperparameters 我使用了以下設定，其餘皆設定為 default 的參數。

max_seq_length=512、learning_rate=3eˆ-5、weight_decay=0、num_train_epochs=1、lr_scheduler_type=linear、per_device_train_batch_size=2、per_device_eval_batch_size=2、gradient_accumulation_steps=4、n_best_size=20、max_answer_length=60。

model 的 optimization alg.是使用 Adam 搭配 linear schedular，learning_rate=3eˆ-5、batch_size=8。



以上兩張圖為此 model 的 training loss 及 EM，可以明顯發現相較於有使用 pretrained weight 的 model 的表現相差許多。
相較於 Q3 的兩張圖，此 model 在一開始時的 loss 高了許多，EM 則低了許多，可從之中看出 pretrained weight 對於 training 的影響力。