Q1: Data processing

1. Tokenizer
   a. Bert tokenizer 為 wordpiece，是類似 byte pair encoding 的一種方式，會將 word 切成 subword，這樣可以避免一些沒看過的字，或是有些相似的字可能會互相影響，像是 love、loved 等等。而兩者差別在於 BPE 是根據出現頻率最高的選擇 subword，而 wordpiece 則是根據最大化機率選擇 subword，演算法如下：
   Step 1: 定義 vocabulary 大小
   Step 2: 將 word 切成 character
   Step 3: 根據 step 2 的資料建立 language model
   Step 4: 選擇能夠增加最大 likelihood 的 subword
   Step 5: 重複 step 4，直到抵達 threshold
2. Answer span
   a. How did you convert the answer span start/end position on characters to position on tokens after BERT tokenization?
   Tokenizer 可以選擇 return_offsets_mapping 會回傳每個 token 對應的(char start, char end)，只要迭代找出 span start 與 char start 相同的位置便為 start position，span end 與 char end 相同的位置便為 end position。
   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?
   對每組 start/end 的配對機率相乘(沒有過 exponential 則為相加)，要將不符合條件的刪除掉，像是 end position < start position 或是 subsentence 比 sentence 長等，找出機率最大的便為最後的結果。

Q2: Modeling with BERTs and their variants

1. BERT
   a. Configuration

      使用 bert-base-chinese


```
{
  "_name_or_path": "bert-base-chinese",
  "architectures": [
    "BertForMultipleChoice"
  ],
  "attention_probs_dropout_prob": 0.1,
  "directionality": "bidi",
  "gradient_checkpointing": false,
  "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": "4.5.0",
  "type_vocab_size": 2,
  "use_cache": true,
  "vocab_size": 21128
}
```



```
{
  "_name_or_path": "bert-base-chinese",
  "architectures": [
    "BertForQuestionAnswering"
  ],
  "attention_probs_dropout_prob": 0.1,
  "directionality": "bidi",
  "gradient_checkpointing": false,
  "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": "4.5.0",
  "type_vocab_size": 2,
  "use_cache": true,
  "vocab_size": 21128
}
```


   b. Performance

      Public data score

      Context selection accuracy: 0.9535

      Question answering EM: 0.78928

      Question answering F1: 0.85528

   c. Loss function

      Cross entropy loss

   d. Training argument

      <u>Context selection:</u>

      optimization algorithm: adamw(lr=3e-5)

      lr scheduler: linear scheduler with warmup, warmup_ratio = 0.1

      batch size: 1

      gradient accumulation step: 64

      <u>Question Answering:</u>

      optimization algorithm: adamw(lr=3e-5)

      lr scheduler: linear scheduler with warmup, warmup_ratio = 0.1

      batch size: 8

      gradient accumulation step: 8

2. RoBERTa-wwm-ext
   a. Configuration

      使用 hfi/chinese-roberta-wwm-ext


```
{
  "_name_or_path": "hfl/chinese-roberta-wwm-ext",
  "architectures": [
    "BertForMultipleChoice"
  ],
  "attention_probs_dropout_prob": 0.1,
  "bos_token_id": 0,
  "directionality": "bidi",
  "eos_token_id": 2,
  "gradient_checkpointing": false,
  "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": 1,
  "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": "4.5.0",
  "type_vocab_size": 2,
  "use_cache": true,
  "vocab_size": 21128
}
```

```
{
  "_name_or_path": "hfl/chinese-roberta-wwm-ext",
  "architectures": [
    "BertForQuestionAnswering"
  ],
  "attention_probs_dropout_prob": 0.1,
  "bos_token_id": 0,
  "directionality": "bidi",
  "eos_token_id": 2,
  "gradient_checkpointing": false,
  "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": 1,
  "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": "4.5.0",
  "type_vocab_size": 2,
  "use_cache": true,
  "vocab_size": 21128
}
```


   e. Performance

      Public data score

      Context selection accuracy: 0.95179

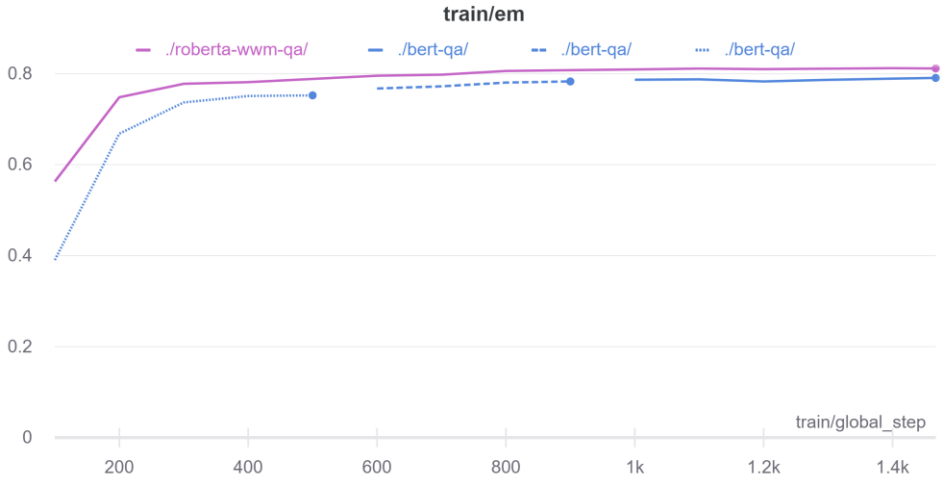
      Question answering EM: 0.81225

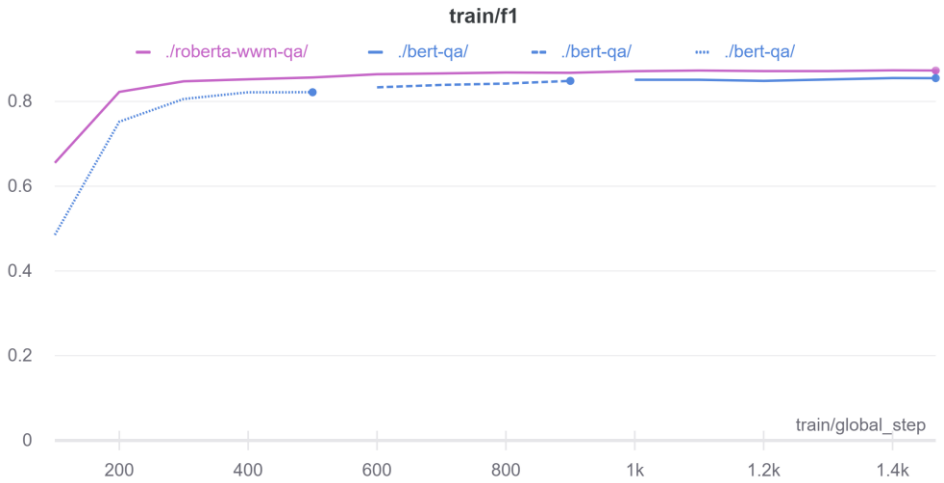
      Question answering F1: 0.87351

   b. Difference

      與 BERT 不一樣的地方在於 Dynamic masking，BERT 在 pre-train 時會先預訂 mask 的位置，在訓練過程中不會改變這些位置，但 RoBERTa 會在一定的時間動態改變 mask 的位置；另外 whole word masking 則是改變針對 subword 做 mask 的方式，如果是同一個 word 有部分的 subword 被 mask 則會將整個 word 都 mask，對於中文，由於最小單位是字，則會先透過一些 pretrained 分詞模型分詞，再做 mask，如果詞有部分字被 mask 則會將整個詞 mask。

Q3: Curves


train/em
./roberta-wwm-qa/
./bert-qa/
./bert-qa/
./bert-qa/
0.8
0.6
0.4
0.2
0
train/global_step
200
400
600
800
1k
1.2k
1.4k


train/f1
./roberta-wwm-qa/
./bert-qa/
./bert-qa/
./bert-qa/
0.8
0.6
0.4
0.2
0
train/global_step
200
400
600
800
1k
1.2k
1.4k

Q4: Pretrained vs Not Pretrained

a. Configuration


```
{
  "_name_or_path": "bert-base-chinese",
  "architectures": [
    "BertForMultipleChoice"
  ],
  "attention_probs_dropout_prob": 0.1,
  "directionality": "bidi",
  "gradient_checkpointing": false,
  "hidden_act": "gelu",
  "hidden_dropout_prob": 0.1,
  "hidden_size": 64,
  "initializer_range": 0.02,
  "intermediate_size": 512,
  "layer_norm_eps": 1e-12,
  "max_position_embeddings": 512,
  "model_type": "bert",
  "num_attention_heads": 4,
  "num_hidden_layers": 2,
  "pad_token_id": 0,
  "pooler_fc_size": 64,
  "pooler_num_attention_heads": 4,
  "pooler_num_fc_layers": 1,
  "pooler_size_per_head": 128,
  "pooler_type": "first_token_transform",
  "position_embedding_type": "absolute",
  "transformers_version": "4.5.0",
  "type_vocab_size": 2,
  "use_cache": true,
  "vocab_size": 21128
}
```

```
{
  "_name_or_path": "bert-base-chinese",
  "architectures": [
    "BertForQuestionAnswering"
  ],
  "attention_probs_dropout_prob": 0.1,
  "directionality": "bidi",
  "gradient_checkpointing": false,
  "hidden_act": "gelu",
  "hidden_dropout_prob": 0.1,
  "hidden_size": 64,
  "initializer_range": 0.02,
  "intermediate_size": 512,
  "layer_norm_eps": 1e-12,
  "max_position_embeddings": 512,
  "model_type": "bert",
  "num_attention_heads": 4,
  "num_hidden_layers": 2,
  "pad_token_id": 0,
  "pooler_fc_size": 64,
  "pooler_num_attention_heads": 4,
  "pooler_num_fc_layers": 1,
  "pooler_size_per_head": 128,
  "pooler_type": "first_token_transform",
  "position_embedding_type": "absolute",
  "transformers_version": "4.5.0",
  "type_vocab_size": 2,
  "use_cache": true,
  "vocab_size": 21128
}
```


減少 head、hidden size、layer 數量

b. Performance

Public data score

Context selection accuracy: 0.36670

Question answering EM: 0.06097

Question answering F1: 0.11008

c. Compare

Loss 需要比較久的時間才會下降，訓練時間久，另外 Loss 雖然有下降但 validation performance 並沒有上升，認為是 transformer 架構還是太巨大，考慮太多 long term 資訊，資料量少的情況容易有 overfitting 發生，需要一定的訓練資料才比較能訓練得起來。