ADL

Homework 2

B07502166 魏子翔

Q1: Data processing.

1. Tokenizer.

   Bert tokenizer爲WordPiece Tokenizer，可以看做是Byte Pair Encoding的變種，其會將句子進行標準化處理，去除非法字元。並且把word切成subword，避免某些相似的字會互相影響。而其與BPE之不同點在於，BPE會選擇出現頻率最高者合併成新的subword，而WordPiece則是根據最大化機率選擇subword。

2. Answer Span.

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

   在tokenize的分割過程中，會在offset_mapping表中記錄其offset量，因此可透過此表還原start/end position在原始context中之位置，

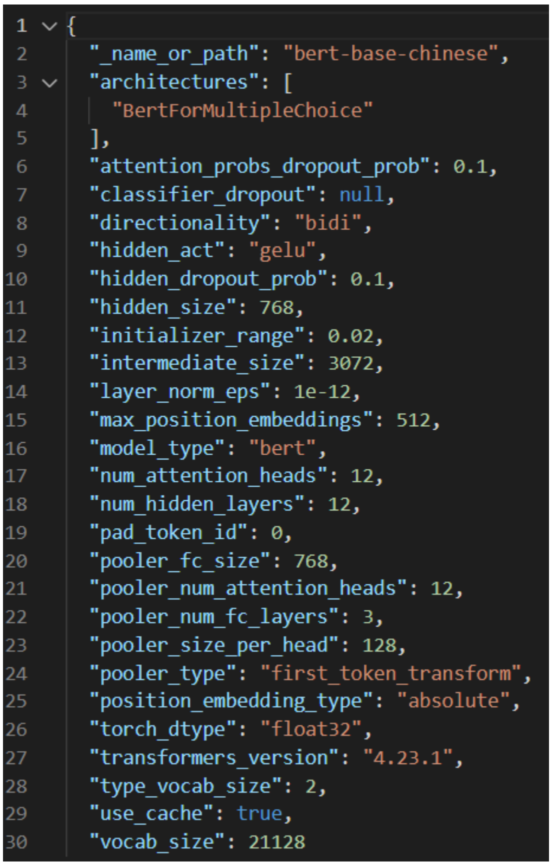
   b. After your model predicts the probability of answer span start/end position, what rules did you apply to determine the final start/end position?

   將每組start/end之機率相加，並將不符合條件者刪除，例如end position < start position，再挑選機率和最大的那組start/end作爲最後結果。

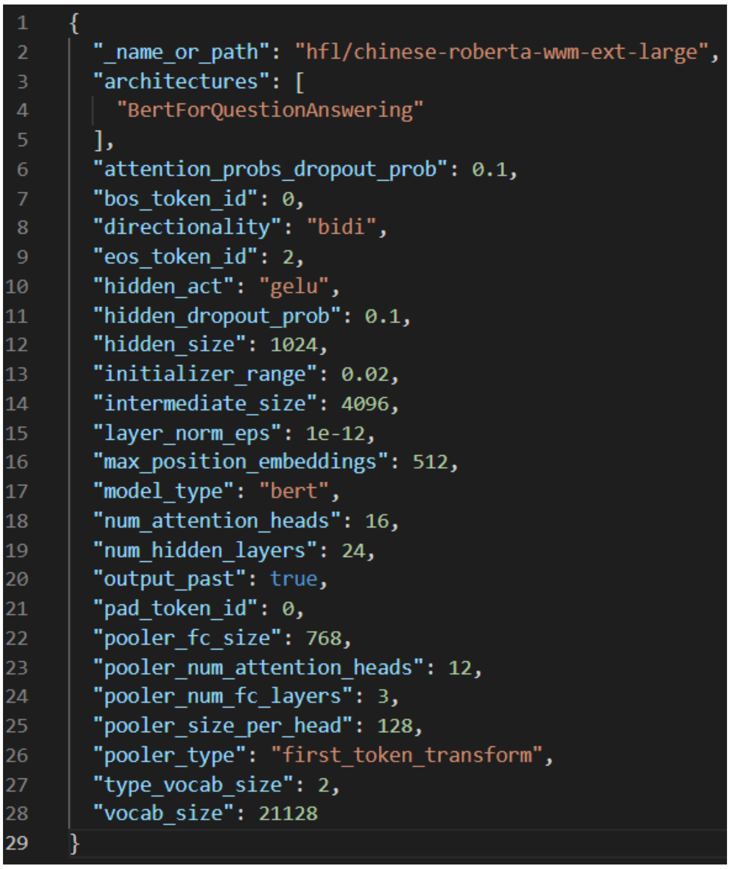
Q2: Modeling with BERTs and their variants.

1. Describe

   a. model configuration



```
{
  "_name_or_path": "bert-base-chinese",
  "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": "4.23.1",
  "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,
  "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",
  "type_vocab_size": 2,
  "vocab_size": 21128
}
```

b. performance

context selection accuracy: 0.956

question answering EM: 0.838

public score: 0.76582

private score: 0.76693

c. loss function

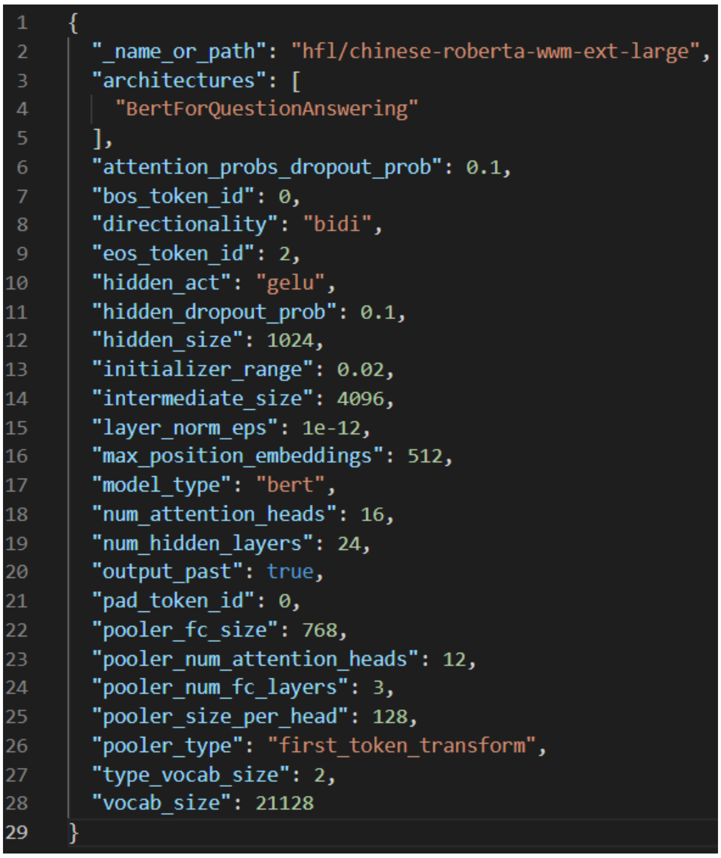
Cross entropy loss

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

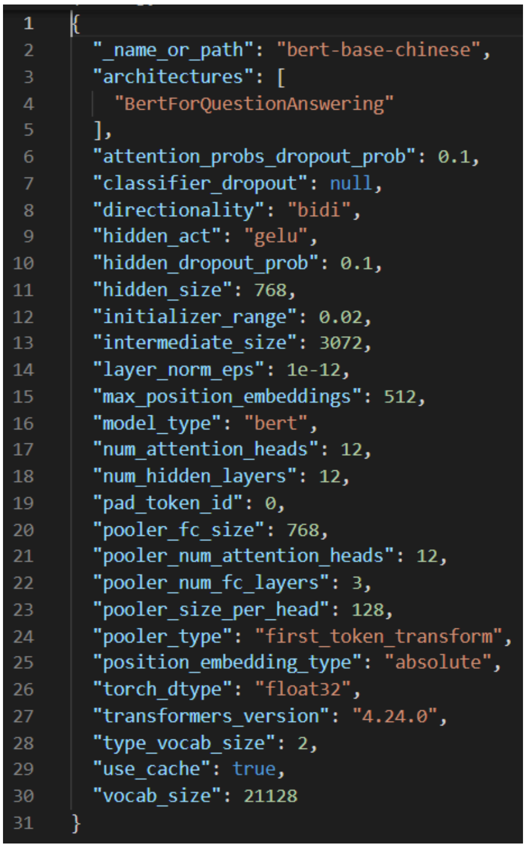
	context selection	question answering
optimizer	AdamW	AdamW
learning rate	3e-5	3e-5
batch size	4	4
num epochs	1	2

2. Try another type of pretrained model and describe.

a. model configuration



```json
{
  "_name_or_path": "hfl/chinese-roberta-wwm-ext-large",
  "architectures": [
    "BertForQuestionAnswering"
  ],
  "attention_probs_dropout_prob": 0.1,
  "bos_token_id": 0,
  "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",
  "type_vocab_size": 2,
  "vocab_size": 21128
}
```



```json
{
  "_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": "4.24.0",
  "type_vocab_size": 2,
  "use_cache": true,
  "vocab_size": 21128
}
```

b. performance

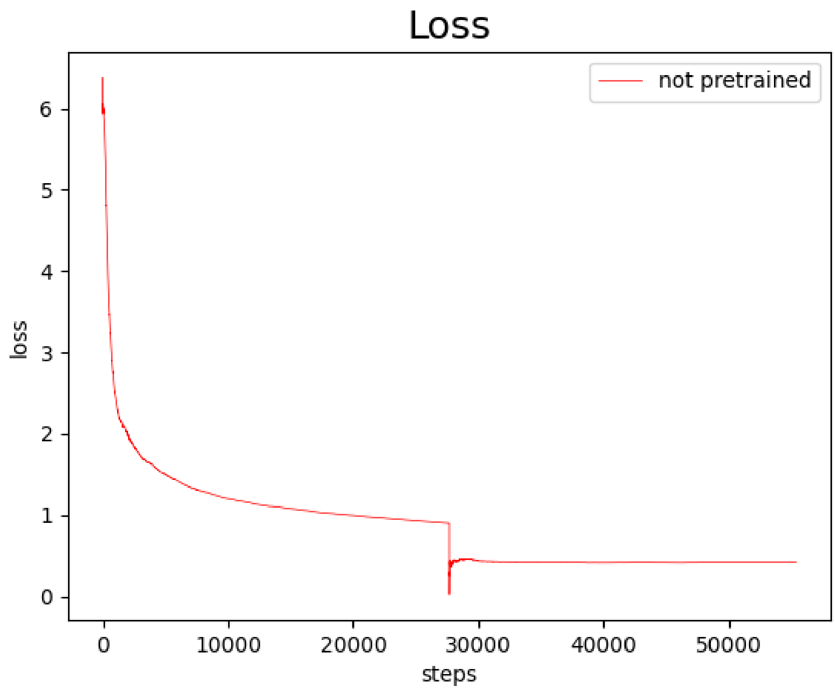
chinese-roberta-wwm-ext-large: 0.838

bert-base-chinese: 0.751

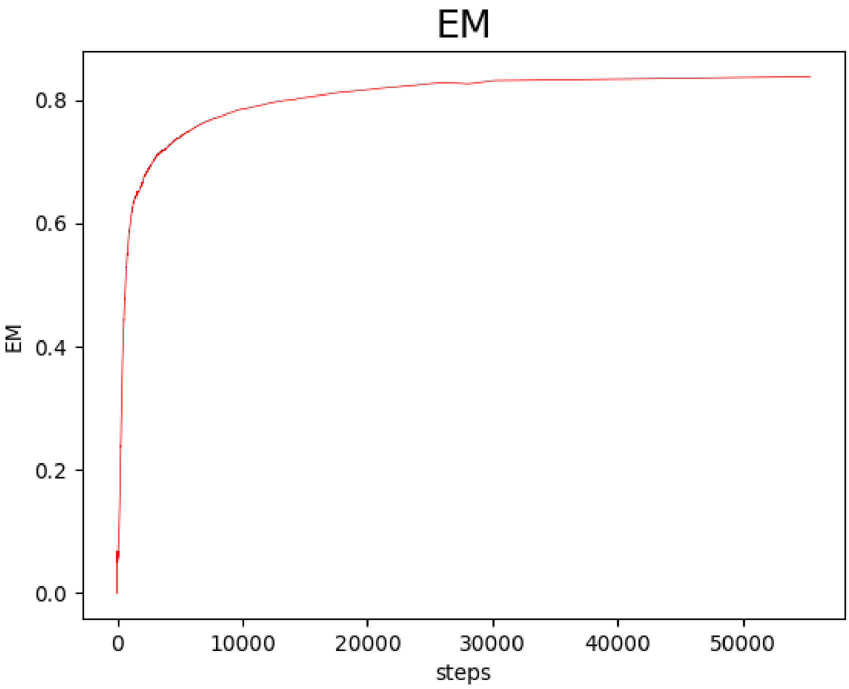
從結果可以看出，roberta-large的表現比bert-base好很多，但其model大小差距較大，訓練時間也有一段差距，因此，我另外嘗試了roberta，其結果爲0.81左右，相較於bert-base之表現亦有大幅度的差距。推測其原因與上課內容相似，是因爲其Dynamic masking的操作與資料量大小，相較於普通的bert，可以獲得更好的表現。

Q3: Curves.

1. Plot the learning curve of your QA model.
   a. Learning curve of loss

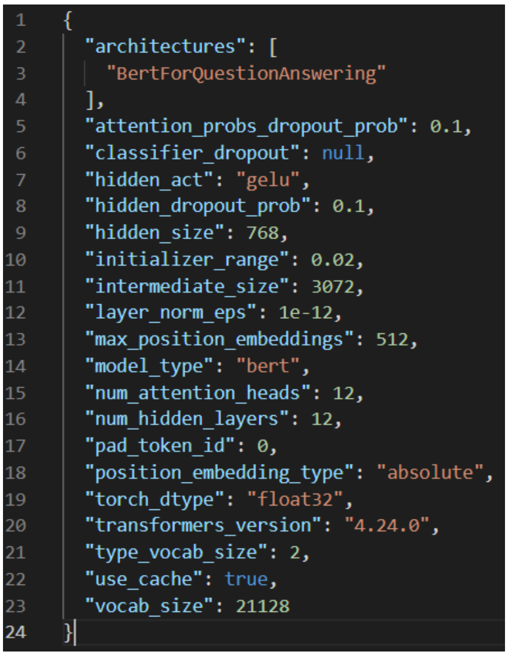


   b. Learning curve of EM

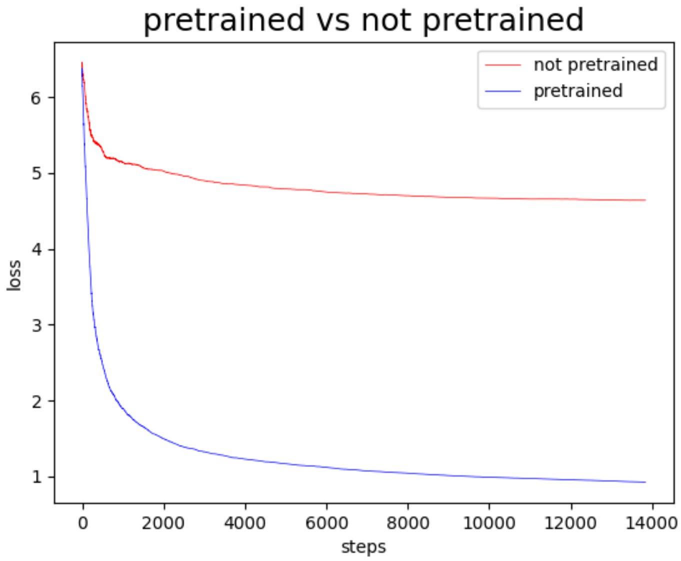


Q4: Pretrained vs Not Pretrained.

以下是QA problem下，有pretrain和無pretrain之比較結果，首先，無pretrain者之model config如左圖所示，其中，optimizer：AdamW, learning rate：3e-5, batch size：4, num epochs：1。 而右圖是此model與使用roberta-wwm-ext pretrained model之performance比較(皆取1 epoch) 可以看出，無pretrained之model loss下降速度非常緩慢，故使用pretrained model可以大幅提高訓練模型之速度，降低所花時間。但not pretrained model之loss還是有在緩步下降，可推測其在消耗大量時間與運算資源後，其表現應可與pretrained model相同。



```
{
  "architectures": [
    "BertForQuestionAnswering"
  ],
  "attention_probs_dropout_prob": 0.1,
  "classifier_dropout": null,
  "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,
  "position_embedding_type": "absolute",
  "torch_dtype": "float32",
  "transformers_version": "4.24.0",
  "type_vocab_size": 2,
  "use_cache": true,
  "vocab_size": 21128
}
```


pretrained vs not pretrained
not pretrained
pretrained
loss
steps