M11152014 蘇珉淙

ADL Homework2 report

Q1: Data processing (2%)

1. Tokenizer (1%):
    a. Describe in detail about the tokenization algorithm you use. You need to explain what it does in your own ways.

做Multiple Choice時，將每個question都複製成四份再各自接上四個可能的paragraphs，再將這四個pairs透過選定的tokenizer轉換成encoding，Tokenizer 則會負責將所有pairs 做padding、truncation成一樣的max_length = 512。

而做Question Answering時會依據Multiple Choice產生的relevant讀取正確的context，接著把question和context接在一起，中間用special token連接，並透過選定的tokenizer轉換成encoding，Tokenizer 一樣會負責將所有pairs做padding、truncation成一樣的max_length = 512，還有透過doc_stride審視整個context，以免因max_length斷句問題導致錯誤的結果。

2. Answer Span (1%):
    a. How did you convert the answer span start/end position on characters to position on tokens after BERT tokenization?

透過tokenizer中的return_offsets_mapping讓我們可以將answer span轉回token的start/end position，若是較長的sequence被拆開也能透過return_overflowing_tokens mapping回去，加上start/end position的判斷即可完成。

    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?

Model的輸出可能不是很按照前後順序，所以需要多去判斷start/end position的順序是否合理，將 start > end的情況去除掉，並找出最佳的pair成為最後的輸出。

Q2: Modeling with BERTs and their variants (4%)

1. Describe (2%)
    a. your model (configuration of the transformer model)

Multiple Choice

```
{
  "_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": "4.22.2",
  "type_vocab_size": 2,
  "use_cache": true,
  "vocab_size": 21128
}
```

Qusetion Answering

```
{
  "_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 your model.

Multiple Choice

```
{
    "epoch": 1.0,
    "eval_accuracy": 0.9617813229560852,
    "eval_loss": 0.13923248648643494,
    "eval_runtime": 122.5067,
    "eval_samples": 3009,
    "eval_samples_per_second": 24.562,
    "eval_steps_per_second": 3.077,
    "train_loss": 0.17984692554227993,
    "train_runtime": 2579.5809,
    "train_samples": 21714,
    "train_samples_per_second": 8.418,
    "train_steps_per_second": 1.052
}
```

Qusetion Answering

```
{
    "eval_EM": 0.7896311066799602
}
```

c. the loss function you used.

Multiple Choice : 使用 Cross Entropy Loss

Qusetion Answering : 使用 Cross Entropy Loss

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

Multiple Choice

Optimizor: AdamW

learning rate: 3e-5

batch size: 2

epoch: 1

Qusetion Answering

Optimizor:AdamW

learning rate: 3e-5

batch size: 2

epoch: 1

2. Try another type of pretrained model and describe (2%)

a. your model

Multiple Choice (使用同樣的pre-trained model)

```
{
  "_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": "4.22.2",
  "type_vocab_size": 2,
  "use_cache": true,
  "vocab_size": 21128
}
```

Qusetion Answering (使用 **hfl/chinese-roberta-wwm-ext**)

```
{
  "_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
}
```

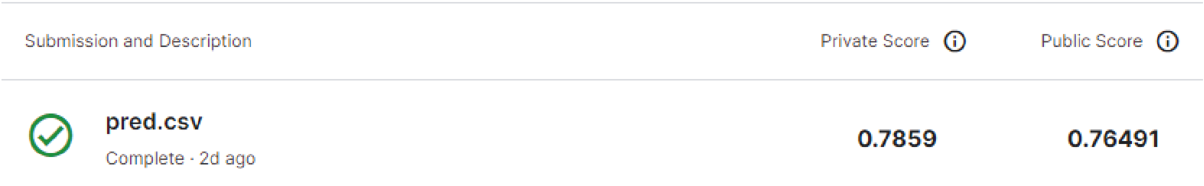
b. performance of your model

Multiple Choice

```
{
   "epoch": 1.0,
   "eval_accuracy": 0.9617813229560852,
   "eval_loss": 0.13923248648643494,
   "eval_runtime": 122.5067,
   "eval_samples": 3009,
   "eval_samples_per_second": 24.562,
   "eval_steps_per_second": 3.077,
   "train_loss": 0.17984692554227993,
   "train_runtime": 2579.5809,
   "train_samples": 21714,
   "train_samples_per_second": 8.418,
   "train_steps_per_second": 1.052
}
```

Qusetion Answering

```
{
   "eval_EM": 0.8148886673313394
}
```



Submission and Description	Private Score ⓘ	Public Score ⓘ
✓ pred.csv Complete · 2d ago	0.7859	0.76491

c. the difference between pretrained model (architecture, pretraining loss, etc.)

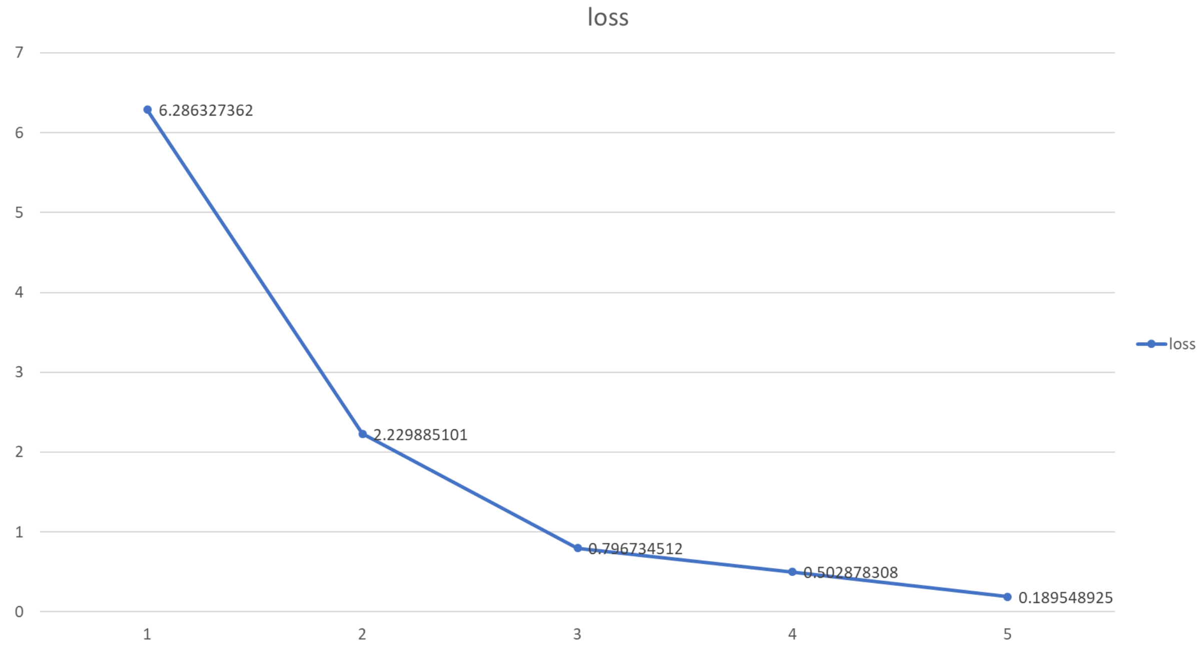
chinese-roberta-wwm-ext-large 改善Bert模型，使用Whole Word Masking在中文分詞效果更佳，再加上large本身就比base大，能夠有效提升準確率。

M11152014 蘇珉淙

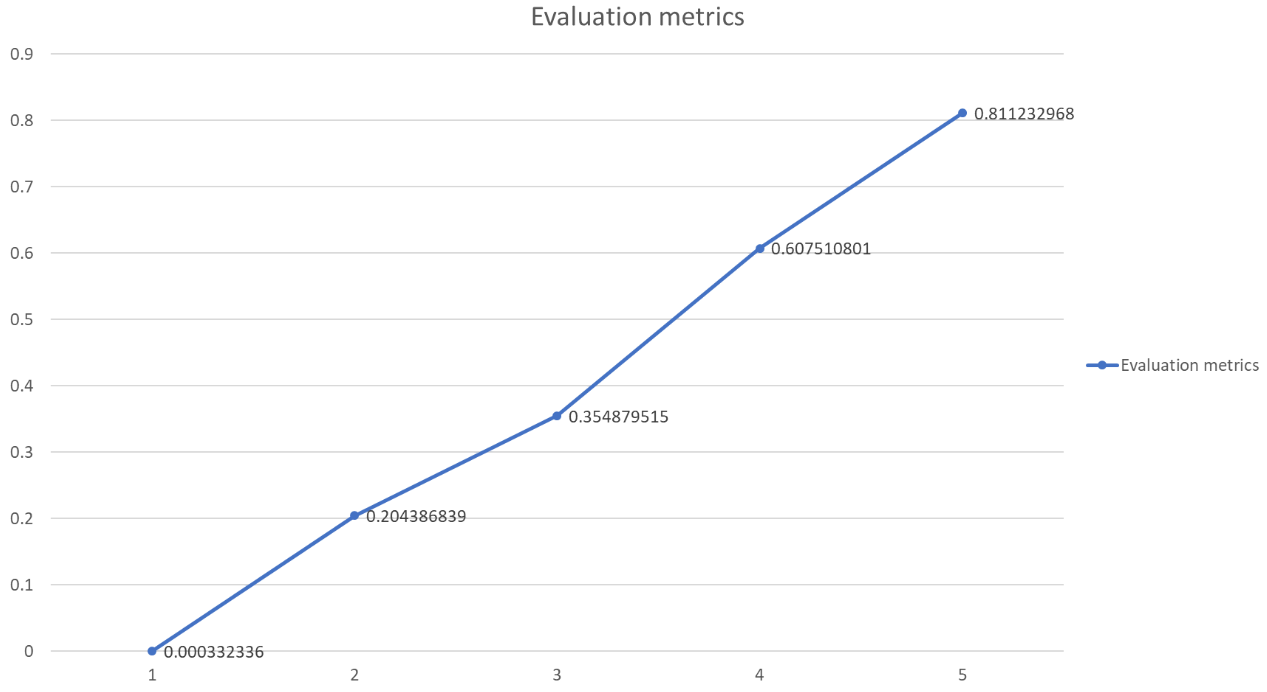
Q3: Curves (1%)

1. Plot the learning curve of your QA model

a. Learning curve of loss (0.5%)



b. Learning curve of EM (0.5%)



Q4: Pretrained vs Not Pretrained (2%)

- Train a transformer model from scratch (without pretrained weights) on the dataset (you can choose either MC or QA)
- Describe
  - The configuration of the model and how do you train this model

    選用QA測試 train from scratch，model_type為bert，tokenizer為bert-base-chinese，讀模型時使用from_config而非from_pretraine。

```
model = AutoModelForQuestionAnswering.from_config(config)
```

```
{
  "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": "4.22.2",
  "type_vocab_size": 2,
  "use_cache": true,
  "vocab_size": 30522
}
```

- the performance of this model v.s. BERT

This model:

Epoch: 3

Evaluation metrics: 0.06347623795280824

loss: 4.27301466464996345

Bert:

Epoch: 1

Evaluation metrics: 0.7896311066799602

loss: 0.843793592755209

Q5: Bonus: HW1 with BERTs (2%)

- Train a BERT-based model on HW1 dataset and describe
  a. your model
  b. performance of your model.
     i. Intent classification (1%)
     ii. Slot tagging (1%)
  c. the loss function you used.
  d. The optimization algorithm (e.g. Adam), learning rate and batch size.