Applied Deep Learning HW2

r11944049 呂瑋浩

October 2022

1 Data processing

我使用BertTokenizerFast為Tokenizer，他是基於Wordpiece實現的，Wordpiece最大的特點是根據likelihood來選擇subword，而非以頻率高低來選擇subword，他可以分為四個步驟：

1. 將word切成character，以中文為例，就會切成一個方塊字
2. 根據1.建立language model
3. 選擇兩個unit合併成subword，並找出提升最大likelihood的subword
4. 重複3直到到達threshold

由於model會使用答案的起始位置與結束位置作為標籤，所以預測的時候也會出現兩個位置表示答案，需要Tokenizer回傳token的mapping，可以藉由return_offsets_mapping達成，預測時，使用model預測的兩個位置，剔除掉不合法的答案(超過長度、開始位置大於結束位置等)，剩下的答案會依照相對應的分數排序後依照mapping重建回char level，並取前n個答案(default n = 20)，取分數最高的non-empty prediction。

2 Modeling with BERTs and their variants

依照助教提供的範例Pipeline，我使用huggingface提供的腳本，並以此測試了兩種模型：bert-base-chinese、hfl/chinese-roberta-wwm-ext。

2.1 bert-base-chinese

```
"_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": "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": "4.23.1",
"type_vocab_size": 2,
"use_cache": true,
"vocab_size": 21128
```

Figure 1: Configurations of Bert-base-chinese

Performance

	Exact match
Context Selection	96.410%
Question Answering	78.132%

Loss function

兩個task的Loss function都使用Cross Entropy，比較不一樣的是Question Answering的標籤是開始位置與結束位置，會分別對兩個位置做Cross Entropy，最終的loss會取平均。

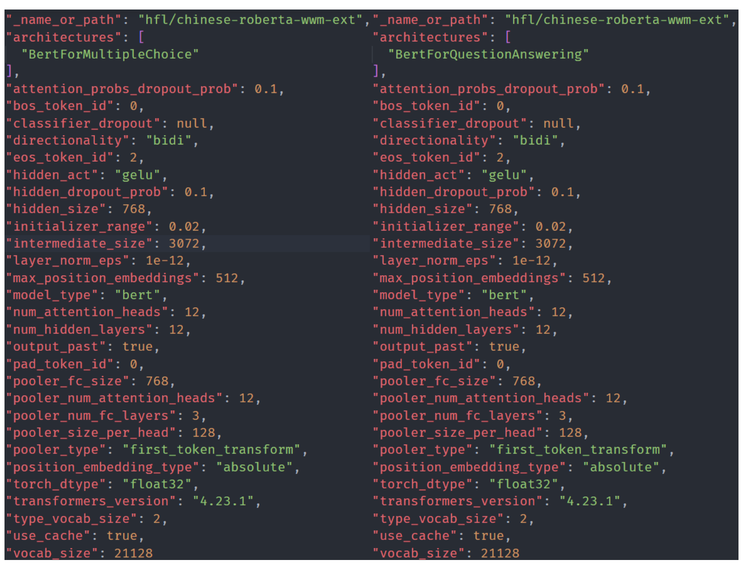
Other training detail

- Context Selection
  - Batch size: 16 (per_gpu_train_batch_size 2 * gradient_accumulation_steps 8)
  - Max_len: 512
  - Num_train_epochs: 3
  - Learning_rate: 3e-5

- Optimizer: AdamW
- Scheduler: Linear decay with warmup

- Question Answering
  - Batch size: 32 (per_gpu_train_batch_size 4 * gradient_accumulation_steps 8)
  - Max_len: 512
  - Num_train_epochs: 3
  - Learning_rate: 3e-5
  - Optimizer: AdamW
  - Scheduler: Linear decay with warmup

2.2 hfl/chinese-roberta-wwm-ext



```
"_name_or_path": "hfl/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": "4.23.1",
"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,
"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": "4.23.1",
"type_vocab_size": 2,
"use_cache": true,
"vocab_size": 21128
```


Figure 2: Configurations of hfl/chinese-roberta-wwm-ext

Performance

	Exact match
Context Selection	96.477%
Question Answering	81.655%

Loss function & Training detail

與Bert-base-chinese相同

Difference

chinese-roberta-wwm-ext與bert-base-chinese主要差別有：

1. 更大的詞彙量
2. Mask方法

在詞彙量的方面，chinese-roberta-wwm-ext的data是取用中文維基百科、其他百科、新聞、問答等數據，整體的詞彙量為5.4B，而bert-base-chinese只有取用中文維基百科，詞彙量為0.4B，chinese-roberta-wwm-ext的資料量遠大於bert-base-chinese。

chinese-roberta-wwm-ext使用了Whole Word Masking，當一個詞彙的一部分被Mask掉，則整個詞彙都會被Mask掉，也就是說在Word piece中，如果「深度學習」這個詞要做Masking，則可能出現[Mask]度學習，而在Whole Word Masking中則會變成[Mask][Mask][Mask][Mask] ，這種Masking方法對中文會更有效率，表現也會更好。Roberta的Masking方法也從靜態的改為動態的。

3 Curves

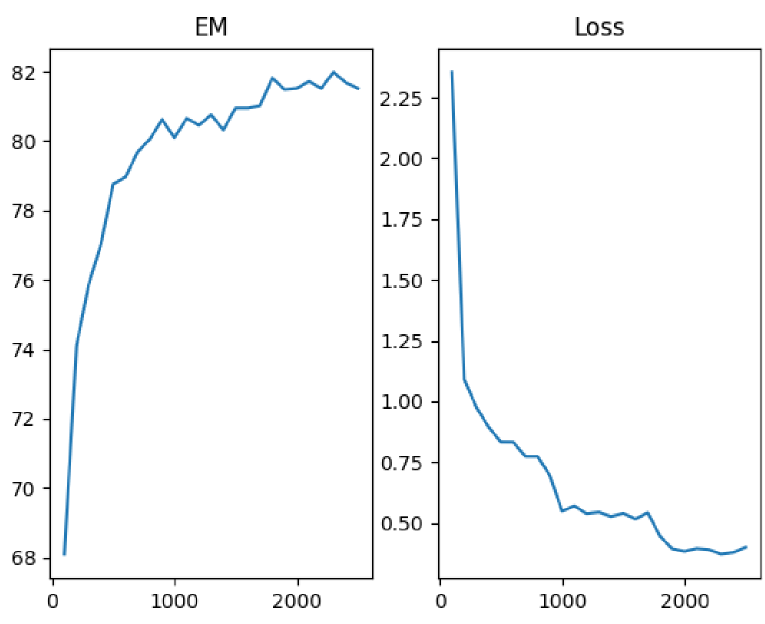


Figure 3: Curve of hfl/chinese-roberta-wwm-ext

每100 step記錄一次loss與EM，總共有2592 step(accumulation steps設定為8)。

4 Pretrained vs Not Pretrained

我在Quesion Answering上實驗hfl/chinese-roberta-wwm-ext有無Pretrained的差異。


```
"_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": "4.23.1",
"type_vocab_size": 2,
"use_cache": true,
"vocab_size": 21128
```


Figure 4: Configurations

兩個model都使用相同的Hyperparameters

- Epochs: 3
- Batch sizes: 32(Per_gpu_train_batch_size 4 ∗ gradient_accumulation_steps 8)
- Learning rate: 3e-5
- Max length: 512

Performance

	Exact match
Non pretrained	4.985%
Pretrained	81.655%

沒有pretrained的情況下，model無法有很好的表現，使用Bert的架構下，用RTX3060跑3個epoch，所需時間為一個小時半，顯示在更大的data與更多epoch下的所需時間相當可觀，並且與pretrain過的model相比，資料量明顯不足，開放train好的模型權重能省掉大量的時間，也能幫助下游任務取得更好的表現，與Computer vision領域中，經常會使用imagenet pretrain過的model可以很快獲得好的表現一樣。

5 Bonus: HW1 with BERTs

使用Bert及Roberta做HW1的兩個task：Intent Classification、Slot Tagging

5.1 Intent Classification

Hyperparameters

- Epochs: 3
- Batch sizes: 8
- Learning rate: 3e-5
- Max length: 512

Performance

	Validation Acc	parameters($\times 10^6$)
BiLSTM	92.600%	15
Bert	95.553%	108
Roberta	96.899%	124

在Intent Classification中，Bert系列的模型很輕鬆地就能超越RNN系列的模型，而與Context Selection、Question Answering兩個task一樣，Roberta的表現也明顯地比Bert優秀。

5.2 Slot Tagging

Hyperparameters

- Epochs: 10
- Batch sizes: 64
- Learning rate: 1e-4

Performance

採用joint accuracy

	Validation Acc	parameters ($\times 10^6$)
BiGRU	82.4%	28
Bert	83.8%	107
Roberta	84.4%	124

在Slot tagging中，Bert系列的模型需要調一下參數才能超越RNN系列，但是在訓練速度上面，Bert比RNN快相當多，slot taggin可以在2分鐘內訓練完成，而RNN受限於模型的特性，需要將近20分鐘才能訓練完成。

Bert系列的模型在HW1的表現比之前的模型有大幅度的提升，但是參數上也明顯增加，所以需要Pretrain過才能在下游任務上有好的表現，Roberta的表現也優於Bert，顯示Roberta不僅是在中文資料上有優勢，在英文資料上也能有優勢。