ADL 2023 Homework 1 Report

學號: B10902116 姓名: 洪子涵 系級: 資工三

Q1

Tokenizer (1%):

Describe in detail about the tokenization algorithm you use. You need to explain what it does in your own ways.

這次使用的 hfl/chinese-roberta-wwm-ext 是中文的 model，對中文用 Character-based tokenizer，每個 Character 都被視為單獨的 token。對其他非中文用的是 WordPiece tokenizer，會學習不同字如何組合會有更大的機率出現，因此 token 中會是不同 Character 組合的 Word。

Answer Span (1%):

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

在 tokenization 後，每個 token 可能會有包含一個到多個 characters，每個 token 的範圍會是 token 包含的第一個 character 到最後一個 character。因此 token 的 start position 對應第一個 character 的 start position，token 的 end position 對應最後一個 character 的 end position。

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

找到每個 token 對應的 start/end position 後，分別選出機率最高的 n_best_size (設定 20)個 start/end position。在每個 start/end position 分別組合成的可能中刪去超過 max_answer_length (設定 30)的組合，並在剩下的組合中選擇 start_logits + end_logits 分數最高的作為答案。

Q2

Describe (2%)

Your model.

將整個過程分為 context selection 和 question answering 兩個部分，先將和題目對應的段落找出來，再進一步在選擇的段落中尋找合適的 answer span。

context selection 使用建議的 multiple-choice model，將四個選項用 bert 處理，將 model 的 logit 視為關聯性，機率最高的代表該選項的 context 和 question 最有相關，因此選擇 logits 最高的作為答案繼續 question answering。

question answering model 將 bert 處理過的 data output 每個 token 計算 start logits 和 end logits，分別代表此 token 作為 answer span 的 start/end position 的機率。將不同 token 的兩個 logits 分別組合後，計算出 start logits + end logits 分

數最高的兩個 token(start/end position 距離不超過指定長度)，將前面 token 的 start position 到後面 token 的 end position 作為最終答案的 answer span。

The loss function you used.

使用的是 Cross-Entropy Loss，計算 model 的預測機率分佈與實際 label 的差異。將實際 label 轉為 one-hot 的向量，實際 label 為 1，其他為 0。

$\hat{y}_i \in \{0, 1\}$代表 i 是不是最相關的選項，$softmax(\mathrm{y}_i)$是選項 i 相關的機率，n 代表選項數量。將 model 的預測機率分佈與實際 label 向量計算每個類別的 loss 後平均得到最後的 loss，最小化這個 loss 以接近實際 label。

$$L = -\sum_{i=0}^{n} \hat{y}_i \ \log(\mathrm{softmax}(y_i))$$

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

1. Architecture: hfl/chinese-roberta-wwm-ext
2. Max sequence length: 512
3. Optimizer: AdamW (weight decay: 0.0)
4. Scheduler: linear
5. Batch size: 4(per_device)*2(gradient_accumulation_steps)
6. Learning rate: 2e-5
7. Epoch: 3

The performance of your model.

Context Selection	eval_accuracy
hfl/chinese-roberta-wwm-ext	0.9521435692921236

Question Answering	eval_exact_match	eval_f1
hfl/chinese-roberta-wwm-ext	81.55533399800598	81.55533399800598

Try another type of pre-trained LMs and describe (2%)

Your model.

流程與 loss function 同上，參數如下。

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

1. Architecture: bert-base-chinese
2. Max sequence length: 512
3. Optimizer: AdamW (weight decay: 0.000008)
4. Scheduler: linear
5. Batch size: 4(per_device)*8(gradient_accumulation_steps)
6. Learning rate: 2e-5
7. Epoch: 8

The performance of your model.

Context Selection	eval_accuracy
bert-base-chinese	0.9564639415088069

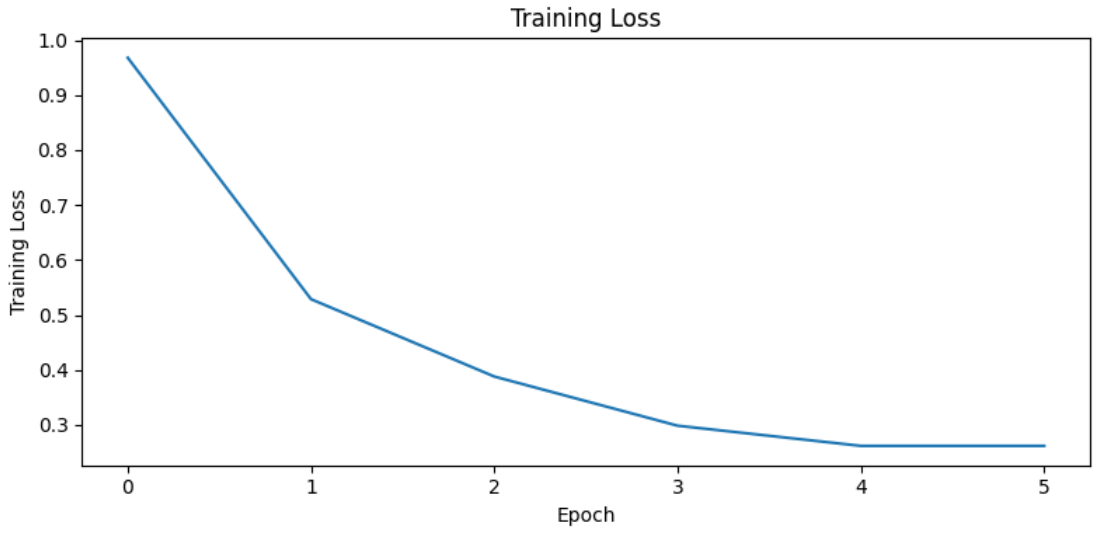
Question Answering	eval_exact_match	eval_f1
bert-base-chinese	79.06281156530409	79.06281156530409

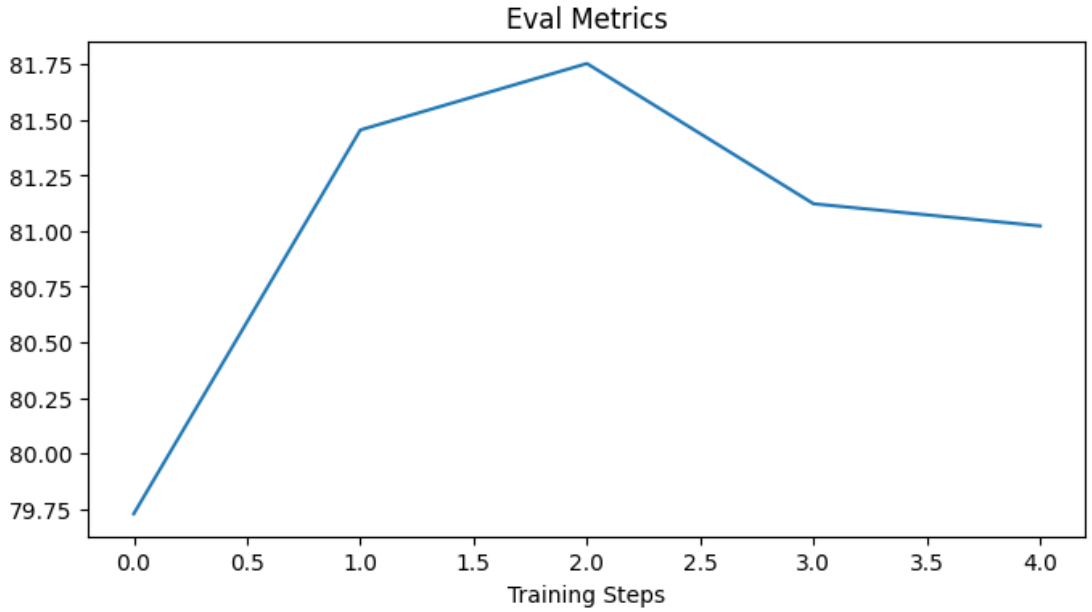
The difference between pre-trained LMs (architecture, pretraining loss, etc.)
bert-base 是最基礎的 bert model，使用 masked 和 next sentence prediction 等技術。
RoBERTa 改成 dynamic mask，代表同一個 sample 在不同的 epoch 時，MASK 掉的 token 有一定機率是不一樣的，並使用更大的 batch size 和更多資料。
Whole Word Masking (wwm) 技術則代表如果一個完整的字的部分被 mask，則同一個字的其他部分也會被 mask。

Context Selection	eval_accuracy
bert-base-chinese	0.9564639415088069
hfl/chinese-roberta-wwm-ext	0.9521435692921236

Question Answering	eval_exact_match	eval_f1
bert-base-chinese	79.06281156530409	79.06281156530409
hfl/chinese-roberta-wwm-ext	81.55533399800598	81.55533399800598

Q3
Plot the learning curve of your span selection (extractive QA) model.
這是使用 hfl/chinese-roberta-wwm-ext 製成的 Learning curve，參數設置同上。





Q4

Train a transformer-based model (you can choose either paragraph selection or span selection) from scratch (i.e. without pretrained weights).

The configuration of the model and how do you train this model (e.g., hyper-parameters).

除了 pretrained weights 之外，所有的參數都和原來一樣。用原來的 code 訓練，只是沒有用 pretrained model。

Context Selection 和 Question Answering 參數皆相同，訓練參數如下：

1. Architecture: hfl/chinese-roberta-wwm-ext
2. Pretrained weights: None
3. Max sequence length: 512
4. Optimizer: AdamW (weight decay: 0.0)
5. Scheduler: linear
6. Batch size: 4(per_device)*2(gradient_accumulation_steps)
7. Learning rate: 2e-5
8. Epoch: 3

Context Selection	eval_accuracy
Train From Scratch	0.5284147557328016
hfl/chinese-roberta-wwm-ext	0.9521435692921236

Question Answering	eval_exact_match	eval_f1
Train From Scratch	7.178464606181455	7.178464606181455
hfl/chinese-roberta-wwm-ext	81.55533399800598	81.55533399800598