Q1: Data processing

Tokenizer

首先先將中文字字串拆成一個個字元，並根據轉換表將字元轉成對應的 token，再來進行 TemplateProcessing，添加 [CLS] 和 [SEP] 特殊標記來讓 transformer 分辨這個是問題還是段落，最後進行 Padding 截到或補足至對應長度，並以 [SEP] 結尾

Answer Span

1.

先利用分詞器輸出的 `offset_mapping` 來執行字到詞的對齊。這個 mapping 提供了每個詞在原始 context 中的字元起始與結束。我們透過遍歷這個映射，尋找第一個其字元起始點大於或等於答案字元起始位置的詞元索引，並將其設為 `token_start`；同時尋找最後一個字元結束點小於或等於答案字元結束位置的詞元索引，並將其設為 `token_end`。

2.

模型輸出了上下文每個詞元的起始和結束分數，接著先枚舉所有有效的區間 (i, j)（其中 $i \leq j$ 且長度 $\leq$ max_answer_length），並計算總分數 $Score(i, j) = \text{Start\_Logit}[i] + \text{End\_Logit}[j]$。之後我們選擇在所有有效組合中具有最高 $Score(i, j)$ 的區間作為最終的答案起始 (token_start) 和結束 (token_end) 位置。另外任何指向問題或特殊詞元的都會被排除。

Q2: Modeling with BERTs and their variants

Describe

Model: bert-base-chinese

Performance:

- eval accuracy: 0.9624459953472915 in paragraph selection
- eval exact match: 80.25922233300099 in question answering

Loss Function: CrossEntropyLoss

Optimizer: Adam, Learning Rate = 3e-5 in both

Batch Size = 2 in both

Epoch = 1 in paragraph selection, Epoch = 3 in question answering.

Try another type of pre-trained LMs and describe

Model: hfl/chinese-lert-base

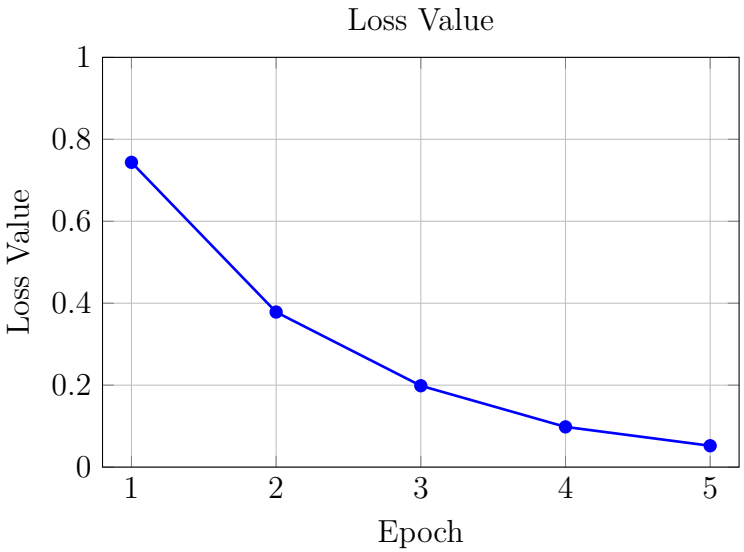
Performance:

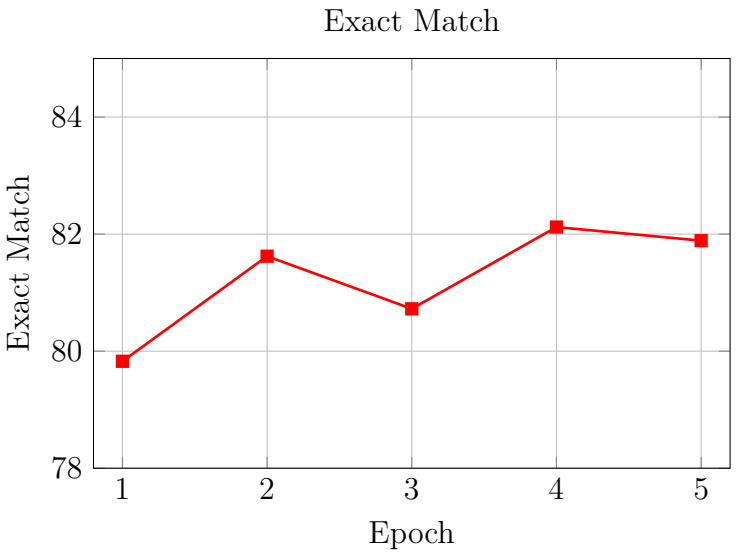
- eval accuracy: 0.9597873047524095 in paragraph selection
- eval exact match: 82.98438019275507 in question answering

這兩種模型的主要差異有二，Lexicon Enhancement 的部分 LERT 在標準 BERT 的基礎上整合了中文詞彙資訊。它在輸入時會同時考慮字元和詞語邊界，將詞彙知識注入到 Transformer 層中。這使其能更準確地理解中文詞義。另外其使用更大的 Pretrain Data，有別於 BERT 僅使用中文維基百科。

Q3: Curves

以下的是使用 validation dataset 測出的 accuracy。另外為方便調用數據，我將 epoch 增加為 5，其餘參數不變。





Q4: Pre-trained vs Not Pre-trained

我嘗試了未 Pretrain 的 Paragraph Selection，仍是使用 BERT，參數均不變，即與 Q2 寫的相同，Tokenizer 也是使用原本的。

試出來的結果是 accuracy=0.37454303755400464，遠遠少於幾乎全對的原模型。

Q5: Bonus

直接將四個文章接在一起，並將其丟給原本的 span 架構使用，只要將 `max_seq_length` 增加到 2048 即可。我並無實作此題，但可以想見這個結果應該是比原本的差，因為模型需要吃更多輸入，會干擾其判斷能力。