ADL hw2 report

學號：R08946006 系級：資料科學學程碩一 姓名：周逸平

Q1: Tokenization (1%)

1-1. Describe in detail how BERT tokenization works.

1-2. What happens when the method is applied on different strings (e.g. Chinese, English or numbers)? Briefly explain your observation.

1-1. BertTokenizer 分為兩個步驟，一為 BasicTokenizer，一為 WordpieceTokenizer，前者可以認定為標準化處理字串，比如 unicode 轉換、標點符號切割、去除非法字元…等等。後者則是將標準化後字串分割成一個一個字元，而注意的是英文中有合成詞的特性，將會被轉換成 ## 的形式，比如說 reading 則會被轉換成 “read”, “##ing”。

1-2. 中文將不考慮詞，而是直接切成單字形式，比如說 ‘你好阿’ 將會被處理為 ‘你’ ‘好’ ‘阿’，而英文則是如 1-1 所述，以詞方式斷句，唯合成詞部分將會以 ## 作為標示。數字的部分，若是數字過長，則同樣以 ## 的方式標示與前字元結合，舉例：

```
a = '169816198419816'
tokenizer.tokenize(a)
 ['169', '##81', '##61', '##98', '##41', '##98', '##16']
```

Q2 : Answer Span Processing (1%)

2-1. How did you convert the answer span start/end position on characters to position on tokens after BERT tokenization?

2-2. After your model predicts the probability of answer span start/end position, what rules did you apply to determine the final start/end position?

2-1. 首先我限制 context 的長度只能為 MAX_CONTEXT_LENGTH，剩餘部分以 PAD 補齊，再與 answer 相連，故輸入的格式為：

CLS, context, PAD, SEP, question, SEP, PAD，若是超出 512 則對 question 做截斷之後再對 answer 做 encode 的動作，並寫一個 function 找出 answer 在 encode 完之後是否有出現在輸入中，若有則回傳 start/end。注意我此處 end 為指針，故若是使用 python list indexing 將會缺尾端一個字。

2-2. 由於我的模型設計上，並不會發生取值取到 question 的問題，故我直接單純的取 輸入[ans_start_pred：ans_end_pred+1] ，若是得到 answerable_pred = 1 但 ans_start_pred > ans_end_pred，則將二者交換。取值完後，若長度 > 30，由於我的 end_loss 較低，故我選擇以 end_loss 作為起點，像左取長度 10 作為輸出，即 ans[-10:]，取 10 的理由是，在 train/valid 中，ans 長度有 95%以上落於 10 之內。但由於這個作法可能會導致選到 special token，故在輸出前去除之。

Q3: Padding and Truncating (1%)

3-1. What is the maximum input token length of bert-base-chinese?

3-2. Describe in detail how you combine context and question to form the input and how you pad or truncate it.

3-1. 512

3-2. 我在 2-1 講完了 XD，詳細的參數設定為

MAX_CONTEXT_LENGTH = 480

MAX_LENGTH = 512

MAX_LENGTH 為輸入總長度

Q4: Model (1%)

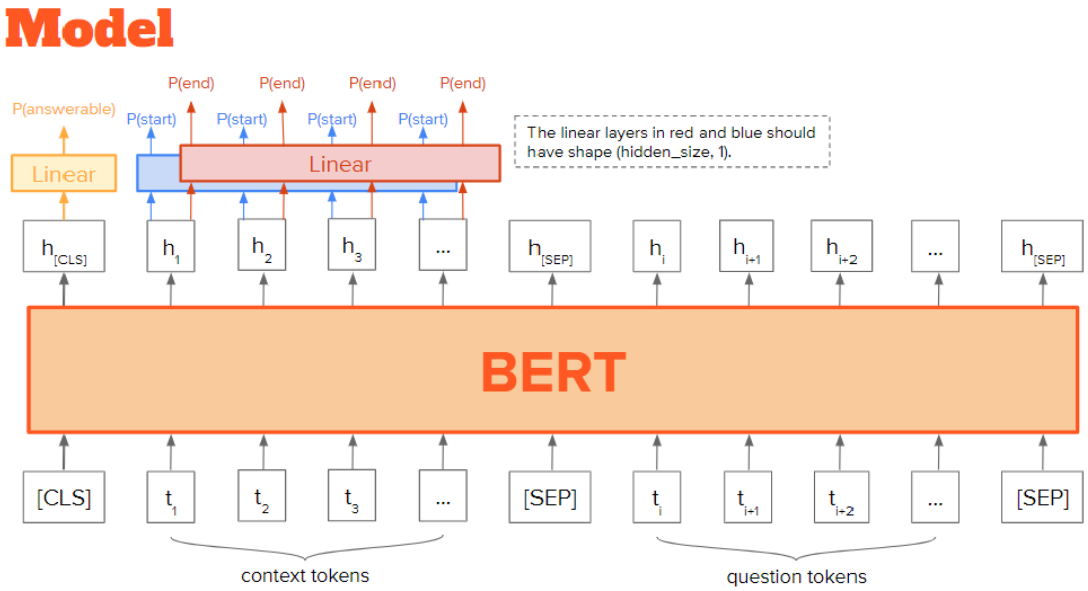
Describe your model in detail.

4-1. How does the model predict if the question is answerable or not?

4-2. How does the model predict the answer span?

4-3. What loss functions did you use?

4-4. What optimization algorithm did you use?



4-1. 基本上與助教模型設計一致，利用 hidden_layer[:,0] 輸入給對應的 linear layer 來得到結果，並以 BCEWithLogitsLoss 計算 loss

4-2. 設計兩組相同的 FC sequential layer 分別為入同樣的輸入，即 hidden_layer[:,1:MAX_CONTEXT_LENGTH+1]，此處這樣設計可以避免採到 question 給出的 output，同時也不會採到起始的 CLE 以及 context 終止符 SEP。另外，於送入 CrossEntropyLoss 之前，將 PAD 的部分 mask 成 –inf。

4-3. BCEWithLogitsLoss : answerable

CrossEntropyLoss : answer start/end

4-4. Transformer 提供之 AdamW, lr = 1e-05, correct_bias = True

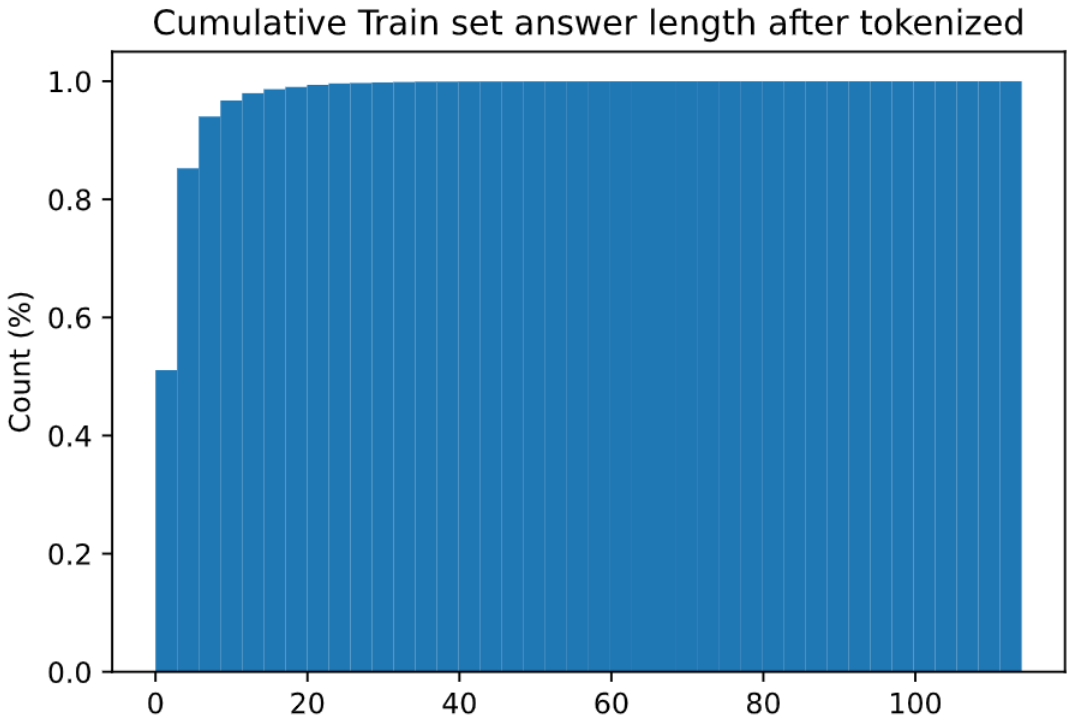
Q5: Answer Length Distribution (2%

)

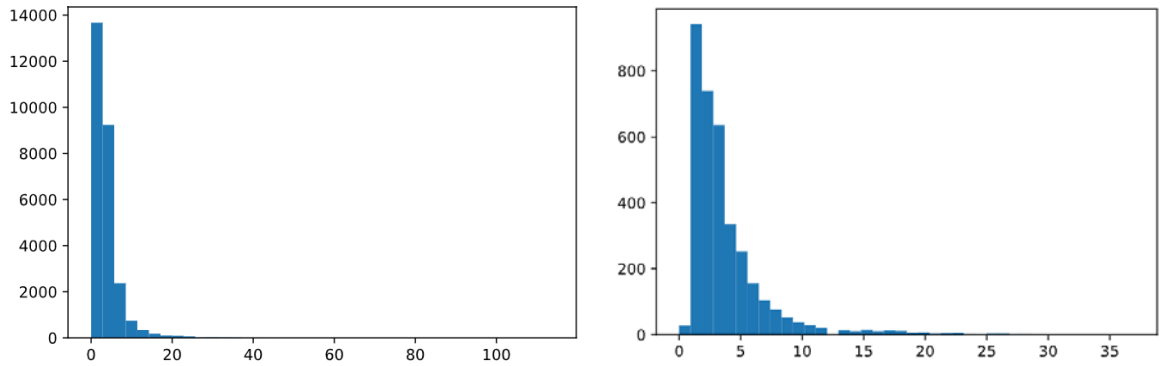
5-1. Plot the cumulative distribution of answer length after tokenization on the training set. (Exclude unanswerable questions.)

5-2. Describe how you can utilize this statistic for finding the answer span given the model output.

5-1.



5-2. 分別觀察在 train 以及 valid 裏頭，其 answer 在 tokenized 之後的長度分別為多少，並繪製成圖：

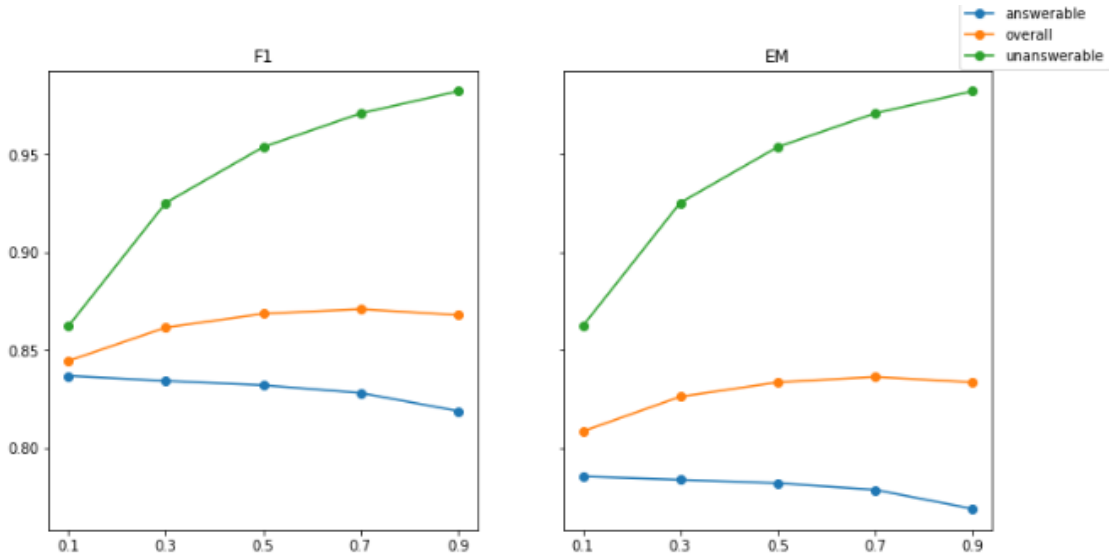


左方為 train 右方為 valid，可以看到在長度 30 之內兩者資料集幾乎都包含了所有的資料，故選擇 30 作為我的 answer_max_length，若 end – st >=30 則輸出為 [st: st+30]

Q6: Answerable Threshold (2%)

6-1. For each question, your model should predict a probability indicating whether it is 6-2. answerable or not. What probability threshold did you use?

Plot the performance (EM and F1) on the development set when the threshold is set to [0.1, 0.3, 0.5, 0.7, 0.9].



Q7: Extractive Summarization (1%)

7-1. You have already trained an extractive summarization model in HW1. No that you are familiar with BERT models, please describe in detail how you can frame the extractive summarization task and use BERT to tackle this task.

7-1. 已知 CLS token 專門用於生成二元預測，且 extractive task 有多個二元預測同時進行，故進行以下步驟。

1. 將句子前後包上 CLS,SEP，如：CLS sent1 SEP CLS sent2 SEP …
2. 將奇數句設定 segments 代號為 0 偶數為 1，以區分不同句子
3. attention 則是維持原設定，PAD = 0
4. 將各 CLS 對應之 last hidden layer 拉出並放入 simple classification layer
5. 對各輸出個別使用 BCE loss 計算並更新參數。