ADL hw2 Report

資管四 連冠旻 b05705009

Q1. Tokenization

中文的bertTokenizer會有21128個token，中文幾乎都是char level的tokenize，有些會有##是接續前一個字的，如果中文的tokenizer碰到大寫英文會變成[UNK]，或是碰到連續的數字比如年份，也會被切在一起，1997就變成'1997'而不是'1"9"9"7'，[UNK]可能會影響performance，所以先設定一個do_lower_case = true來避免。

Q2. Answer Span Processing

我幾乎沒用到原本給的start，大略就是contexts, text tokenize完，拿token去比對，我是用比較遜的方式，在context裡面找到第一個答案就return位置了，但也有可能後面會有重複的，不過看起來這方法的分數能過baseline。

而在predict的時候，先把token id的index拿出來，然後用index去token id tensor裡面找，找到之後再切出來，切出來的就是答案的token id list，那在用tokenizer.convert_ids_to_tokens轉成token，接著做一些後處理把空白拿掉、#拿掉等等。

Q3. Padding and Truncating

bert-base-chinese最大input token length是512，我用pad_to_max_length=true進行padding，然後truncate是先算出questions長度+[CLS]+2[SEP]是多少，512再減去這些就是context可以丟進去的長度，如果太長就truncate掉超過的。

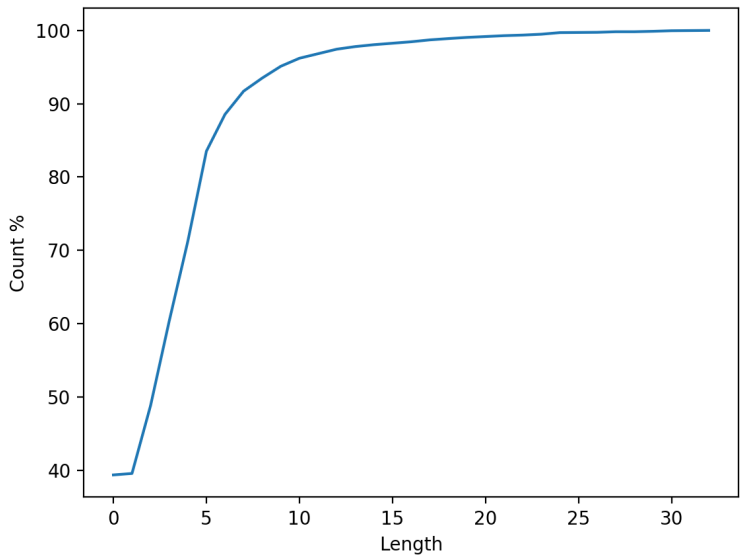
Q4. Model

假如算出來的start end順序有問題，或是長度太長(>30)的話我就會當作unanswerable。

我的model是用bertforquestionanswering，所以train的時候會塞我的start_positions, end_positions兩個label，eval的時候他就會自動回傳兩個值，那也就是predict出來的start index & end index。

我是直接用pytorch套件的loss.backward()，然後optimizer是用AdamW配上lr=e-6

Q5. Answer Length Distribution



由此看見在1~4的長度是指數型的上升，所以可以說是答案通常都很短，大約一個詞就結束了。

Q6. Answerable Thershold

我是用start end的index和長度去做是否answerable，所以沒有一個threshold

Q7. Extractive Summarization

利用bert-extractive-summarizer，model = Summarizer()，label就丟答案進去。