r08944035 呂翊愷

ADL Homework #2

Q1: Tokenization

BERT對於中文字的tokenize，會將每個字分開，接著再找出每個字在BERT的vocabulary中的id，將每個字換成id。如果遇到BERT的vocabulary中沒有的字，BERT會將它視為[UNK]，將該字換成[UNK]的id。

對於數字的tokenize，BERT會先看這串數字有沒有在vocabulary中，如果在，就直接換成id，如果不在，BERT會先將這串數字中的最後一個數字分開，再看前面的數字在不在vocabulary，如果在，就直接把前面的數字換成id，如果不在，BERT一樣會把將這串數字中的最後一個數字分開繼續找，以此類推直到找到為止。一旦找到在vocabulary中的數字，就會開始找後面原本被捨棄掉的數字是不是在vocabulary中，步驟和前面的相同。而BERT為了能知道後面的數字是接續前面的數字，會將後面的數字加上”##”表示該字是接續前面的。

對於英文的tokenize，BERT會先根據sentence中的空格將每個word分開，找出每個word有沒有在vocabulary中，如果在，就直接換成id，如果不在，BERT會先將這個word中的最後一個letter分開，再看前面的word在不在vocabulary，如果在，就直接把前面的word換成id，如果不在，BERT一樣會把將這串word中的最後一個letter分開繼續找，以此類推直到找到為止。一旦找到在vocabulary中的word，就會開始找後面原本被捨棄掉的word是不是在vocabulary中，步驟和前面的相同。而BERT為了能知道後面的word是接續前面的word，會將後面的數字加上”##”表示該word是接續前面的。

Q2: Answer Span Processing

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

我會將answer前面的context做tokenization，產生出來tokenize的長度就是answer的start position。接著我再將answer做tokenization，得到answer做tokenize後的長度，加上answer的start position就是answer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?

對於model預測出來start/end的機率，我會將非context的部分先mask掉，設為-inf，然後再找出最大值的index當作是start/end的position。如果start跟end的position相差大於30，或start的position大於end的position，這兩種情況我認為代表model無法預測出一個好的答

案，因此我就直接輸出答案是空字串。

Q3: Padding and Truncating

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

512個token。

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

我將question的長度上限設為59個token，也就是question的長度超過的話，我就會truncate到59個token。由於整個input的長度上限是512，所以context的長度上限會是512 - question的長度 - 3，超過的話就會truncate，之所以會需要再減掉3個token是因為這3個token是用來保留給[CLS], [SEP], [SEP]這些特殊符號用的。

由於同一個batch裡面的每個input的長度可能會不同，輸入model的長度又必須一致，因此我使用`torch.nn.utils.rnn.pad_sequence`這個函式，它能將同一個batch中較短的input全部用0來pad到最長的input的長度。

Q4: Model

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

將BERT的output[1]經過linear變成1維的vector，再設一個threshold = 0.6，如果此數大於等於threshold就是answerable，小於threshold就是unanswerable。

- How does the model predict the answer span?

將BERT的output[0]分別經過2個linear得到2個512維的vector，這2個vector每一維都代表該token是start/end的機率，此時我再將這兩個vector中數字最大的2個index分別當作start/end的position。

- What loss functions did you use?

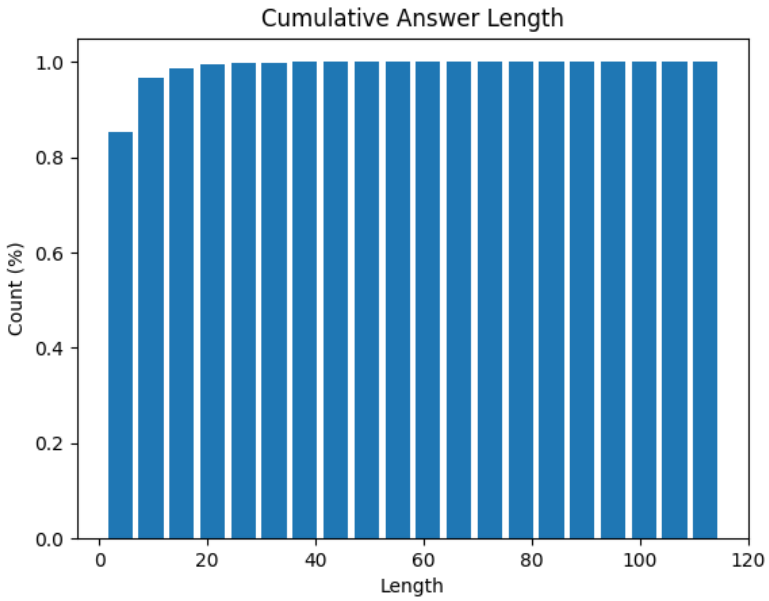
預測answerable的部分是用Binary Cross Entropy，預測start/end position是用Cross Entropy。

- What optimization algorithm did you use?

使用Adam optimizer，learning rate設為9e-6。

Q5: Answer Length Distribution

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



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

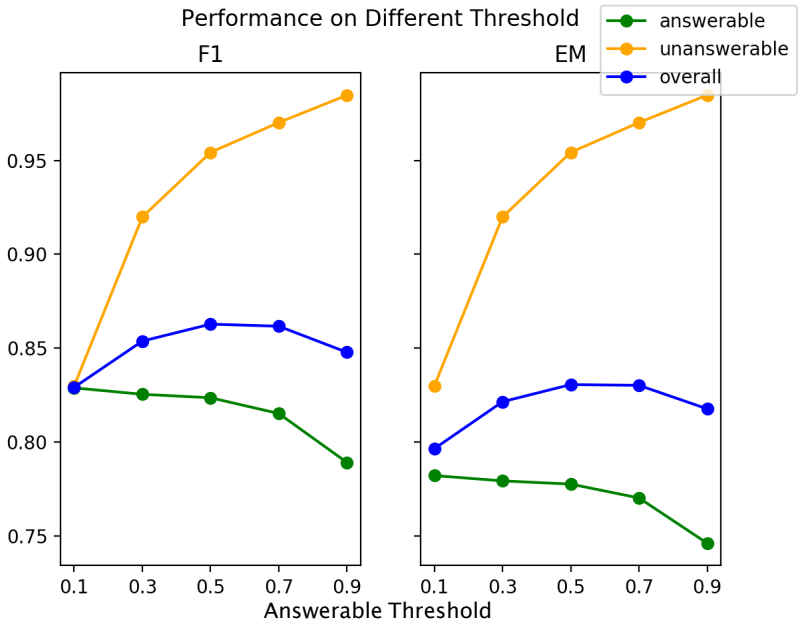
由上圖可以得知，在長度30的時候，cumulative的比例已經快要接近100%了，代表答案的長度幾乎都是在30以下，所以在預測答案的時候，如果得到的答案長度超過30的話，可以將它捨棄，因為答案長度超過30的比例很少。

Q6: Answerable Threshold

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

我的threshold設為0.6。

- 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

- 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.

將整篇文章做tokenize之後輸入到BERT，再接一層Linear layer，變成一個512 * 1維的向量，此向量是每個token是否為summary的機率，將這個向量經過sigmoid後，把文章中每個句子所屬的token是summary的機率相加之後除以該句子的長度，就可以得到每個句子是否為summary的機率，此時取機率最大的句子當作該篇文章的extractive summary即可。