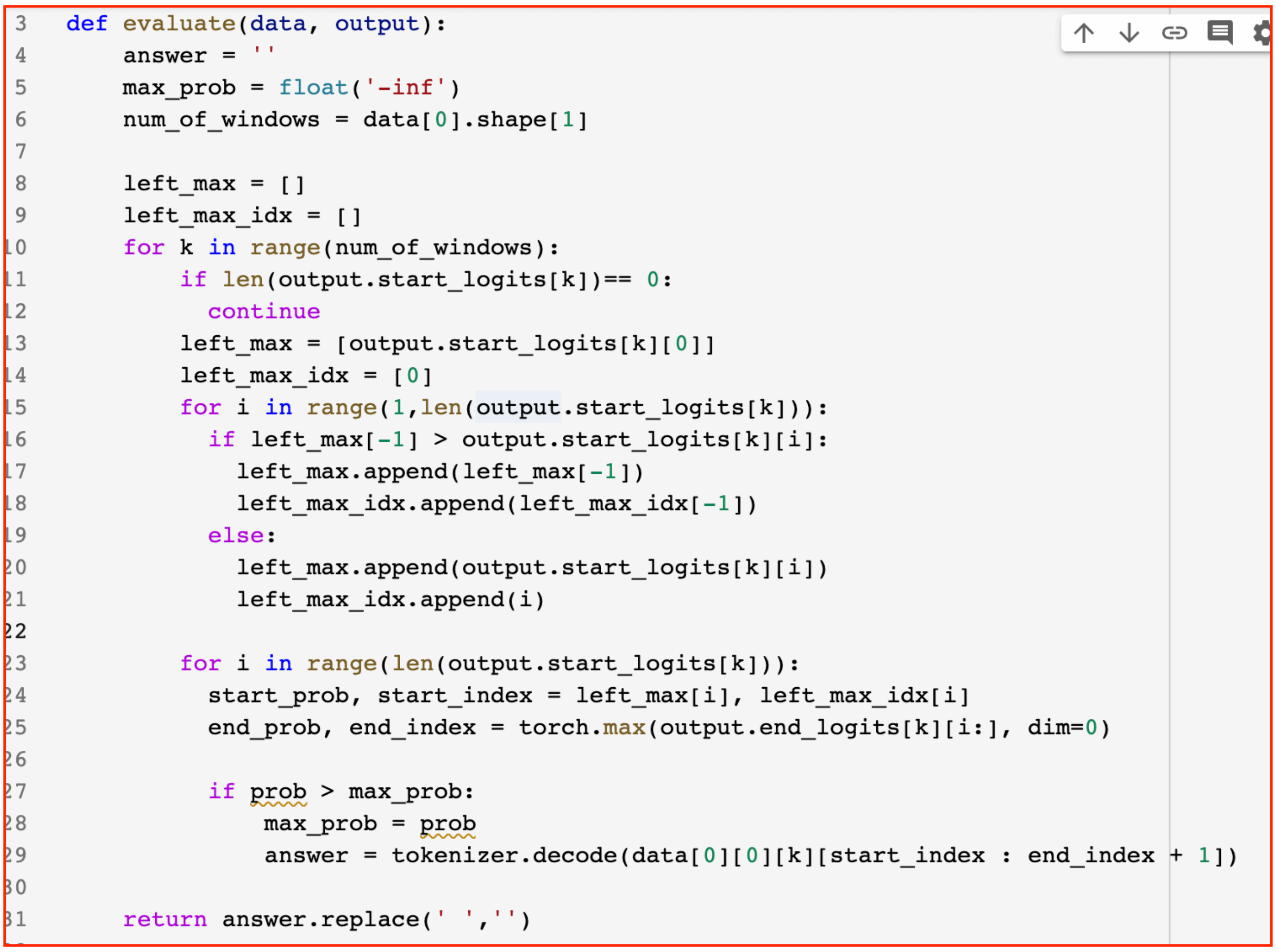
1. (2%) After your model predicts the probability of answer span start/end position, what rules did you apply to determine the final start/end position? (the rules you applied must be different from the sample code)

Ans:
排除掉開始位置＞絕對位置的這種可能，並且取總和最大。具體來說，就是對於每一個i<j，都試著求出prob(start_from_i && end_at_j)的機率，並且求出最大。

然而，如果直接用for-loop-i和for-loop-j這種雙層迴圈來求解，複雜度多了一個order，需要數十個小時才能得出結果。 //時間複雜度O(nn), 空間複雜度O(1)

因此，我們改用loop up table的方式，先記錄start_from_j這邊的最大值，再搭配end_at_j來輸出解答。 //時間複雜度O(n), 空間複雜度O(n)



```python
def evaluate(data, output):
    answer = ''
    max_prob = float('-inf')
    num_of_windows = data[0].shape[1]

    left_max = []
    left_max_idx = []
    for k in range(num_of_windows):
        if len(output.start_logits[k])== 0:
          continue
        left_max = [output.start_logits[k][0]]
        left_max_idx = [0]
        for i in range(1,len(output.start_logits[k])):
          if left_max[-1] > output.start_logits[k][i]:
            left_max.append(left_max[-1])
            left_max_idx.append(left_max_idx[-1])
          else:
            left_max.append(output.start_logits[k][i])
            left_max_idx.append(i)

        for i in range(len(output.start_logits[k])):
          start_prob, start_index = left_max[i], left_max_idx[i]
          end_prob, end_index = torch.max(output.end_logits[k][i:], dim=0)

          if prob > max_prob:
              max_prob = prob
              answer = tokenizer.decode(data[0][0][k][start_index : end_index + 1])

    return answer.replace(' ','')
```

2. (2%) Try another type of pretrained model which can be found in huggingface's Model Hub (e.g. BERT -> BERT-wwm-ext, or BERT -> RoBERTa), and describe

(a) the pretrained model you used
(b) performance of the pretrained model you used
(c) the difference between BERT and the pretrained model you used (architecture, pretraining loss, etc.)

Answer：
pretrain模型名稱：
"hfl/chinese-macbert-base"

使用後的效能：
0.69584->0.74263

和BERT的差別：
參考說明網站"https://huggingface.co/hfl/chinese-macbert-base"，macbert是使用了MLM進行pretrain，是一種微調過的BERT優化版本。換言之，他減輕了pretaining和finetuning之間的差異程度。

另一方面，macbert也使用了這些的技術—Whole Word Masking (WWM), N-gram masking, Sentence-Order Prediction (SOP)。這也都造成了bert與macbert之間的區別性。

更多的資訊，可以參考這篇論文
"Cui, Yiming, et al. "Revisiting pre-trained models for Chinese natural language processing." *arXiv preprint arXiv:2004.13922* (2020)."