AI Final Project Report Group 09

蘇時頤、羅方妤、呂賴臻柔

1. Introduce your model, what are advantages of your model.

資料前處理：

(1)以路徑、檔名將data 放進來(一次讀一個題目，避免程式跑不動)變成一個list

(2)將1~21句都放入list中(pre_text:1~20句題目、target_text:21句題目、choice:備選答案)

(3)將target_text的"XXXXX"改成"_"、”AAAAA”忽略、choice選項以”|”分開

(4)將pre_text與target_text合併為text，若text長度超過2000，則將問題刪減為剩下10~20句

(5)以"[MASK]"表示"_"，未來bert將判別此位置為要預測的克漏字，將答案寫入

Training Model---Bert：

採用bert的pre train model做字的挑選預測

(1)計算model

(2)將輸入的資料換成一個個token，再轉為id，最後將id轉為tensor

(3)若RuntimeError則會跳出系統，不繼續做判斷

(4)若無(3)的問題則將所有tensor放進model做預測，得出最後的結果

(5)將最後得出的結果寫入target_text的"_"中

可能結果與解決方法：

- 成功預測且正確的結果

```
predicted_token:
 said
After predicting:
 21 `` They are very old friends of our family , my dear , that 's all , '' _said_ the king timidly .
The Answer is:
 8   said
--- 51.77058482170105 seconds ---
```

- 預測出結果但非備選答案

```
predicted_token:
 woman
After predicting:
 21 said the Fingers ; but they had to hold her tight while the _woman_ dropped some sealing-wax on the nee
dle and stuck it in the front of her dress .
C:\Users\matteosoo\AppData\Local\conda\conda\envs\BERT\lib\site-packages\gensim\matutils.py:737: FutureWarn
ing: Conversion of the second argument of issubdtype from `int` to `np.signedinteger` is deprecated. In fut
ure, it will be treated as `np.int32 == np.dtype(int).type`.
  if np.issubdtype(vec.dtype, np.int):
[-0.022314636036753654, 0.04800107702612877, 0.17405052483081818, 0.08134152740240097, 0.049388010054826736
, 0.10359489917755127, 0.10726013779640198, 0.0918027013540268, 0.14868773519992828, 0.1866447478532791]
Most similar answer is:
 10   waist   0.1866447478532791
check1:  9
check1:  9
The Answer is:
 10   waist
--- 179.44548511505127 seconds ---
```

  - 解決方法：用word2vec的pre train model(使用word2vec-google-news-300)比對答案與備選答案兩者的關聯，使用最相關的備選答案作為答案

- 長度過長無法餵進BERT model作mask預測的RuntimeError情況
  - 解決方法：若字母>2000則將一開始測試的20句刪減為10~20句，再做預測(這次比賽沒有遇到這個問題)
- word2vec的pre train model訓練的字不夠多，造成沒有預測答案或備選答案的字，無法判斷相關性

```
  File "C:\Users\matteosoo\AppData\Local\conda\conda\envs\BERT\lib\site-packages\gensim\models\keyedvectors
.py", line 452, in word_vec
    raise KeyError("word '%s' not in vocabulary" % word)
KeyError: "word 'clover-leaf' not in vocabulary"
```

- 解決方法：將這個找不到的答案的相關性設為最低，使這個找不到的答案不可能被選中，再從其他備選答案找出相關性最高者做為答案(這次比賽沒有遇到這個問題)

最終結果：準確率0.8(上述可能發生的問題在此次比賽並無發生)

2. What is the weakness of your model.

- bert這個model無法讓我們一次輸入10筆data，我們必須將data分為十筆，才不會跑不動
- data太長不能跑:我們將21個問題何在一起成為list，導致list太長會不能跑(約>2000個字母會失敗)
  - 遇到問題:因刪減預測值，可能導致預測結果不準確
- 可能出現word2vec的pre train model沒有備選答案或預測答案的字，而無答案
  - 遇到問題：只能將找不到的答案設為最不相關，但根本上沒有解決這個問題

3. How to improve it

- **bert這個model無法讓我們一次輸入10筆data的問題**
  bert model All 是用Cloud TPU 64GB RAM，但若使用12GB - 16GB RAM 的GPU，又用同樣的hyperparameters，就有可能有out-of-memory 的問題，以現行常見的GPU硬體設備來說，google的BERT官網有提到會新增Gradient accumulation及Gradient checkpointing去避免記憶體不足的問題；另外，google所提供的雲端colab也有提供TPU的資源去使用，這個也是避免這樣的狀況的其一解決之道。
- **data太長不跑的問題**
  model中直接將問題刪減為10~20句，等於沒有考慮任何因素直接砍掉一半的data，可能造成error rate很高，依據Machine Learning 學到的Schapire's boosting 發想，可以將training set分成多組小的data，每組data都要有部分的文字是與前一組重複的，每組data放進主程式跑，最後用一個master classifier決定哪個答案得到最多的votes，就以他當最終答案，如下面Pseudo code。

```
1
2   if len(pre_text)>2000:
3       length = pre_text//250
4       last = len(pre_text) - 250 * length
5       for n in range(length - 3):
6           text = pre_text[0+250 * n:1000+250 * n] + target_text
7           主程式
8       text = pre_text[-1000:] + target_text
9       主程式
10  else:
11      test = pre_test +target_text
12      主程式
13  排序出現最多次的answer，選那個，若沒有，random
14
```

- **word2vec的pre train model沒有備選答案或預測答案的字，而無答案的問題**
  model中我們直接將沒有在pre train model的字相關性設最低，讓這個答案完全不可能被選中，但這樣其實也是不可靠的，等於這個答案是無效的，若是能夠在

pre train model中找到這個字的相似字來替代，可能可以提高一些正確率，又或者因為無法決策，有是或不是這兩個選項，所以取個丟銅板丟到正面（二選一）的期望值0.5，或是因為十個選項都有可能為答案，所以將他的機率設為0.1，可能會提升正確率，未來可以自己train model，讓答案更符合備選答案。

4. What is the work division of each team member

工作內容	工作分配
發想程式演算法	全體組員
撰寫程式	
Debug	
撰寫報告	

5. Reference data

- BERT model Pytorch實作參考：
  https://github.com/huggingface/pytorch-pretrained-BERT
- BERT架構的attention配置及memory問題參照：
  https://www.jiqizhixin.com/articles/2018-11-01-9
- BERT詞向量的feature extract參考：
  (但沒有成功實作應用，因為發現相近詞向量周圍的詞句意義相差太多)
  https://zhuanlan.zhihu.com/p/50773178
- word2vec找相近詞：
  https://github.com/RaRe-Technologies/gensim-data
- google官方BERT github
  https://github.com/google-research/bert

6. Any feedback

感謝老師與助教認真的教學，這學期的課程受益良多，雖然這次我們並沒有建立出真正training的model，不過藉由bert的練習，讓我們經由資料讀取、model的建立到最後比對出最接近的結果對於這次的project有更深入的了解，也對AI的領域產生更多的興趣，未來可以再多方學習，更深入了解AI的世界。