Q1 Data processing :

1. Tokenizer

助教給的 train.json 檔如下圖:

```
{
  "id": "ab39567999fd376480ac3076904e598e",
  "question": "舍本和誰的數據能推算出連星的恆星的質量？",
  "paragraphs": [
    5234,
    6952,
    8264,
    836,
    92,
    2018
  ],
  "relevant": 836,
  "answers": [
    {
      "text": "斯特魯維",
      "start": 108
    }
  ]
},
```

我將所有的 paragraphs 接在一起變成一整大篇的文章，並且重新計算 answer start 的位置，也因為這樣做的關係我避免了做分類的任務(判斷 relevant)， 當初會想這樣做是因為覺得如果把功課分成兩個任務去做，這樣整體準確率不就會變成 accuracy of stage 1 * accuracy of stage 2 ，這樣可能導致我的準確率較低(但結果並不是我想的那樣)， 而雖然我們文章長度變為原本的 6~7 倍，但是因為 bert model 有使用 doc_stride 因此應該不會降低模型準確率太多，於是就這樣做下去了。

```
  "paragraphs": [
   {
    "context": "所有的恆星，有生之年的絕大部分時間都是主序星，主要是燃燒氫元素，經由核融合產生氦。然而，不同質量的恆星在其演化階段有著截然不同的性質，大質量恆星不僅最終的命運和低質量恆星不同，它們的亮度和對周遭環境的衝擊也不同。因此，天文學家經常以質量將恆星分成不同的群組。非常低質量的恆星是質量少於0.5太陽質量的恆星，不會演化進入漸近巨星分支，但是會直接成為白矮星。低質量恆星是質量超過0.5太陽質量，但未超過1.8-2.2太陽質量的恆星，會演化進入漸近巨星分支，在那裏演化出簡併的氦核。中等質量恆星會經歷氦融合和演化出簡併的碳-氧核。大質量恆星的質量至少是7-10太陽質量，但也可能低至5或6太陽質量。這些恆星在生命的後期經過碳融合，並以核心坍縮的超新星爆炸結束一生。心理學是否為自然科學的範圍，目前也尚存爭議，一般較廣為接受的說法是心理學同時包含在自然科學與社會科學的範疇之中。自然科學的根本目的在於尋找隱藏在自然現象背後的規律，但是自然科學的工作尚不包括研究為什麼會存在這些規律。自然科學認為超自然的、隨意的和自相矛盾的現象是不存在的。自然科學的最重要的兩個支柱是觀察和邏輯推理。由對自然的觀察和邏輯推理自然科學可以引導出大自然中的規律。假如觀察的現象與規律的預言不同，那麼要麼是因為觀察中有錯誤，要麼是因為至此為止被認為是正確的規律是錯誤的。一個超自然因素是不存在的。這是一個長久以來就存在的假設，大多數的恆星都是長期處在特定重力場的多星或聯星系統。特別是許多大質量的歐和逼型恆星，有80%被認為是多星系統的一部分。然而，質量越低的恆星，單獨存在的比例顯然越高，只有25%的紅矮星被發現有伴星。因為85%的恆星是紅矮星，所以在銀河系內多數的恆星都是單獨誕生的。恆星在宇宙中的分布是不均勻的，並且通常都是成群的與星際間的氣體、塵埃一起存在於星系中。一個典型的星系擁有數千億顆的恆星，而在可觀測宇宙中的星系數量超過一千億個。2010年對恆星數量的估計是在可觀測宇宙中有3000垓顆。儘管人們往往認為恆星僅存在於星系中，但星系際的恆星已經被發現了。在19世紀雙星觀測所獲得的成就使重要性也增加了。在1834年，白塞爾觀測到天狼星自行的變化，因而推測有一顆隱藏的伴星；愛德華·皮克林在1899年觀測開陽週期性分裂的光譜線時發現第一顆光譜雙星，週期是104天。天文學家斯特魯維和舍本·衛斯里·伯納姆仔細的觀察和收集了許多聯星的資料，使得可以從被確定的軌道要素推算出恆星的質量。第一個獲得解答的是1827年由菲利克斯·薩瓦里透過望遠鏡的觀測得到的聯星軌道。對恆星的科學研究在20世紀獲得快速的進展，相片成為天文學上很有價值的工具。卡爾·史瓦西發現經由比較視星等和攝影星等的差別，可以得到恆星的顏色和它的溫度。1921年，光電光度計的發展可以在不同的波長間隔上非常精密的測量星等。阿爾伯特·邁克生在虎克望遠鏡第一次使用干涉儀測量出恆星的直徑。超新星爆炸會使這顆恆星的大部分物質都飛散出去。剩下的就是中子星，或是在質量最大恆星就會形成黑洞。在中子星內的物質是中子簡併物質，和一種可能存在核心但極不穩定的簡併物質，垮克物質。物質在黑洞核心所處在的狀態是迄今仍不了解的。垂死恆星拋出去的外層物質包括一些重元素，可能恆星形成的世代交替中成為新恆星的原料。這些重元素可以形成岩石的行星。從超新星和大恆星的恆星風拋出的物質在星際物質的構成中扮演著重要的角色。除了單獨的恆星之外，恆星系統可以是兩顆或更多的恆星受到重力的約束而在軌道上互繞著。最普通的恆星系統就是聯星，但是也發現有三顆或更多恆星的系統。而因為軌道要穩定的緣故，這些恆星系統經常會形成階級制度的共軌恆星。也存在著更大的、被稱為星團的集團。他們的範圍從只有幾十顆恆星，鬆散的星協，到龐大的擁有數十萬顆恆星，稱為球狀星團的集團。1930年，印度物理學家蘇布拉馬尼揚·錢德拉塞卡根據廣義相對論計算出質量大於1.4倍太陽質量的非轉動星體會因重力塌縮成為電子簡併態。愛丁頓雖然在理論上支持黑洞存在的可能性，但同時認為錢德拉塞卡的觀點事實上不能成立，他認為「應當有某種自然定律阻止恆星出現這種荒唐的行為」。當時的物理學家如波耳、亨利·羅素等人都贊同錢德拉塞卡的理論，但出於愛丁頓聲望的原因，他們並沒有公開對錢德拉塞卡表示支持。不過從某種意義上說，愛丁頓也是正確的，當恆星質量大於錢德拉塞卡極限後，確實仍然會有中子簡併壓力阻止恆星繼續塌縮。到了1939年，美國物理學家羅伯特·歐本海默等人推算了這種情形的恆星質量上限，這個極限被稱作托爾曼-歐本海默-沃爾科夫極限。當今的天體物理學家普遍認為，除非有如未知的夸克簡併壓力一類因素的存在，質量大於托爾曼-歐本海默-沃爾科夫極限的恆星將最終會塌縮為錢德拉塞卡所預言的黑洞。即使如此，史瓦西解作為當時能夠描述黑洞行為的唯一精確解，由於具有一些讓人看似不大優美的性質以及難以與實驗觀測相聯繫，一直沒有進入主流物理學研究的視野，關於黑洞的理論乃至整個廣義相對論領域的研究由此擱置了二十年之久。",
     "qas": [
      {
       "question": "舍本和誰的數據能推算出連星的恆星的質量？",
       "id": "ab39567999fd376480ac3076904e598e",
       "answers": [
        {
         "text": "斯特魯維",
         "answer_start": 977
        }
```

而在丟資料進 bert model 的方式如下圖顯示

```
tokens: [CLS] 平 安 京 的 設 計 是 採 用 中 國 什 麼 時 代 的 京 都 格 局 ？ [SEP] 斯 就 已 經 有 關 於 音 樂 的 研 究 ， 古 希 臘 的 阿 夫 洛 斯 管 及 里 拉 琴 也 漸 漸 發 展 為 
```

也就是 [CLS] question [SEP] paragraph context [SEP]，每個字會去 bert 的 vocab.txt 檔(如下圖) 找尋對應的 id ，[CLS] 起頭符號、[SEP] 分割符號以及 [UNK] 未知詞。

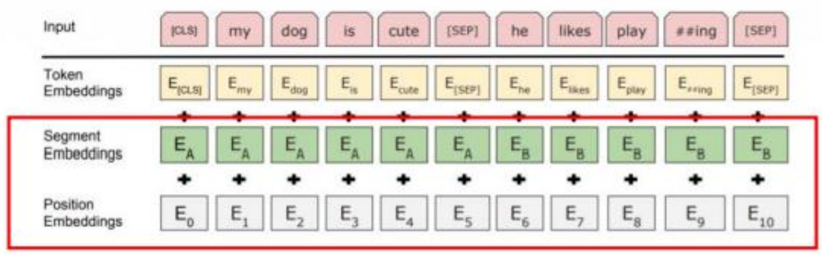
```
975  偏
976  偕
977  做
978  停
979  健
980  側
981  偵
982  偶
983  偷
984  僂
985  偽
986  償
987  傀
988  傅
989  傍
990  傑
991  傘
```

由於使用 doc_stride 的關係，且 bert input size 最大只有 512 (我最大設 200 個字，因為要在 8G 的空間上 train RoBERTa large)故需要把文章拆成許多小文章，這樣就必須要做(token to original map)的動作，input id 為把 input 每個字對應到 vocab.txt 的索引，segment ids 就是把 question 用 0 表示 ，paragraph context 用 1 表示，下圖可見:

```
token_to_orig_map: 23:1597 24:1598 25:1599 26:1600 27:1601 28:1602 29:1603 30:1604 31:1605 32:1606 33:1607 34:1608 35:1609 36:1610 37:1611 38:1612 39:1613 40
token_is_max_context: 23:False 24:False 25:False 26:False 27:False 28:False 29:False 30:False 31:False 32:False 33:False 34:False 35:False 36:False 37:False
input_ids: 101 2398 2128 776 4638 6257 6243 3221 2967 4500 704 1751 784 7938 3229 807 4638 776 6963 3419 2229 8043 102 3172 2218 2347 5195 3300 7302 3176 750
input_mask: 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
segment_ids: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
```

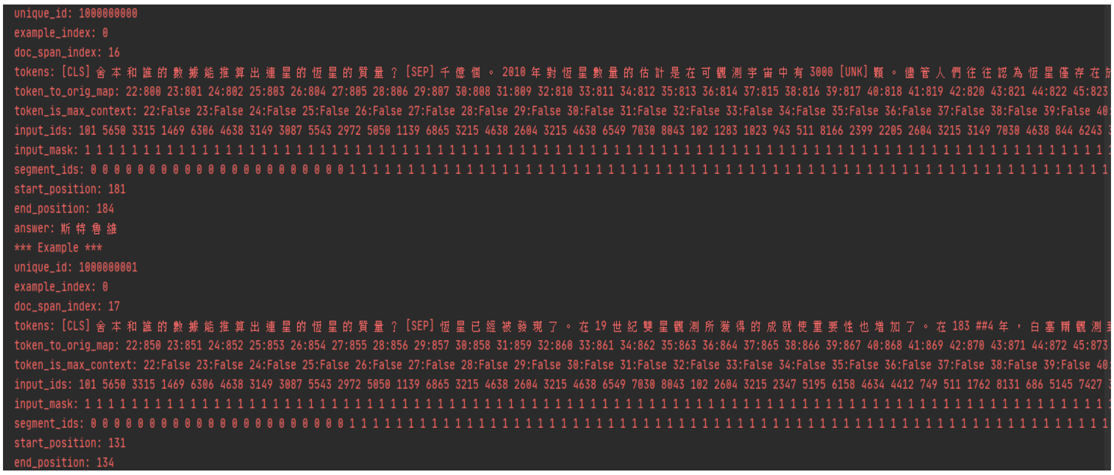
而標準的 bert input 有三層:

Token Embeddings、Segment Embeddings、Position Embeddings



2. Answer Span

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



Answer 並不會出現在所有文章中(因為我將所有文章合成一個大文章)，所以在做 training data 的時候，我只會將滑窗有節錄到答案的文章段落做成 training data 如上圖所示 start_position 為當前滑窗 答案:斯特魯維 用 str.find 去找並返回當前此答案的位置。而 end_position 就會是 start_position+len(answer)。

(b)After your model predicts the probability of answer span start/end position, what rules did you apply to determine the final start/end position?

"hidden size":1024 出來外面會接上一層 nn.linear(1024, 2)，2 代表一個是 start position 另一個是 end position，最後會接上 softmax 來做 normalize 機率，最後再選出機率最大的 start position 和 end position。

Model= BertForQuestionAnswering.from_pretrained()

sequence_output=model(input_ids, segment_ids, input_mask, start_positions, end_positions)

sequence_output .shape = [batch_size, max seq len, 1024]

self.qa_outputs = nn.Sequential(nn.Linear(config.hidden_size, 2))

logits = self.qa_outputs(sequence_output)

logits.size=[batch_size, max seq len, 2]

Q2: Modeling with BERTs and their variants

1. Describe (2%)
   a. your model

      Chinese-roberta-wwm-ext-large
   b. performance of your model.

      EM:76.6

      F1:83

c. the loss function you used.

Loss 分成 start logit 以及 end logit 兩個部分:

loss_fct = CrossEntropyLoss()

start_loss = loss_fct(start_logits, start_positions)

end_loss = loss_fct(end_logits, end_positions)

total_loss = (start_loss + end_loss) / 2

d. The optimization algorithm (e.g. Adam), learning rate and batch size.

Implements BERT version of Adam algorithm with weight decay fix

BertAdam: lr=3e-5

warmup_proportion=0.1(前面 10 趴的 data 會使用較小的 LR 訓練)

b1=0.9, b2=0.98, e=1e-6, weight_decay=0.01

2. Try another type of pretrained model and describe (2%)

a. your model

Chinese-bert-base

b. performance of your model.

EM: 68

F1: 71

c. the difference between pretrained model

RoBERTa 採用的是 dynamic masking 方式，且在 optimizer adam 超參數 β2 改為 0.98 這樣一來使的 training 在大的 batch size 比較穩定，在訓練數據上讓 model 看了更多的資料，包含了 BookCorpus, CC-News, OpenWebText, Stories，這也是 RoBERTa 比 Bert 還要強大的主要原因。

在 model config 上得差異如下:

Chinese-bert-base:


```json
{
  "attention_probs_dropout_prob": 0.1,
  "directionality": "bidi",
  "hidden_act": "gelu",
  "hidden_dropout_prob": 0.1,
  "hidden_size": 768,
  "initializer_range": 0.02,
  "intermediate_size": 3072,
  "max_position_embeddings": 512,
  "num_attention_heads": 12,
  "num_hidden_layers": 12,
  "pooler_fc_size": 768,
  "pooler_num_attention_heads": 12,
  "pooler_num_fc_layers": 3,
  "pooler_size_per_head": 128,
  "pooler_type": "first_token_transform",
  "type_vocab_size": 2,
  "vocab_size": 21128
}
```

Chinese-roberta-wwm-ext-large:


```json
{
  "architectures": [
    "BertForMaskedLM"
  ],
  "attention_probs_dropout_prob": 0.1,
  "bos_token_id": 0,
  "directionality": "bidi",
  "eos_token_id": 2,
  "hidden_act": "gelu",
  "hidden_dropout_prob": 0.1,
  "hidden_size": 1024,
  "initializer_range": 0.02,
  "intermediate_size": 4096,
  "layer_norm_eps": 1e-12,
  "max_position_embeddings": 512,
  "model_type": "bert",
  "num_attention_heads": 16,
  "num_hidden_layers": 24,
  "output_past": true,
  "pad_token_id": 1,
  "pooler_fc_size": 768,
  "pooler_num_attention_heads": 12,
  "pooler_num_fc_layers": 3,
  "pooler_size_per_head": 128,
  "pooler_type": "first_token_transform",
  "type_vocab_size": 2,
  "vocab_size": 21128
}
```


d. The optimization algorithm (e.g. Adam), learning rate and batch size.

Implements BERT version of Adam algorithm with weight decay fix

BertAdam: lr=3e-5

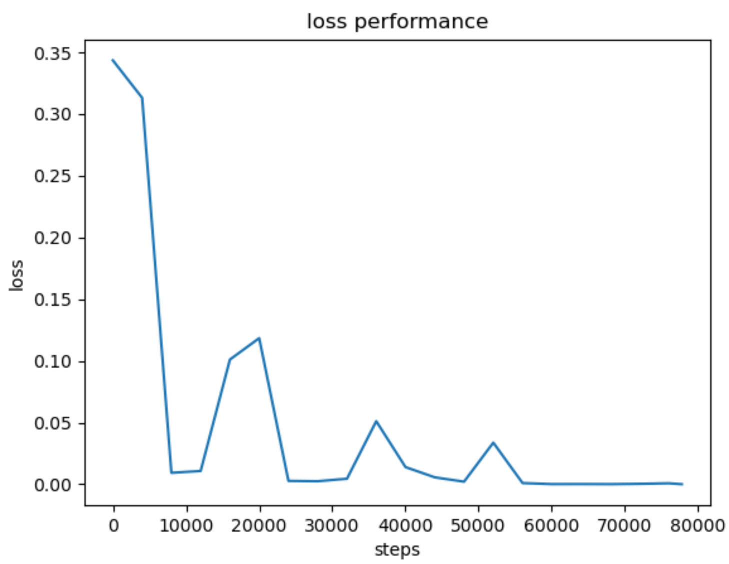
warmup_proportion=0.1(前面 10 趴的 data 會使用較小的 LR 訓練)

b1=0.9, b2=0.999, e=1e-6, weight_decay=0.01

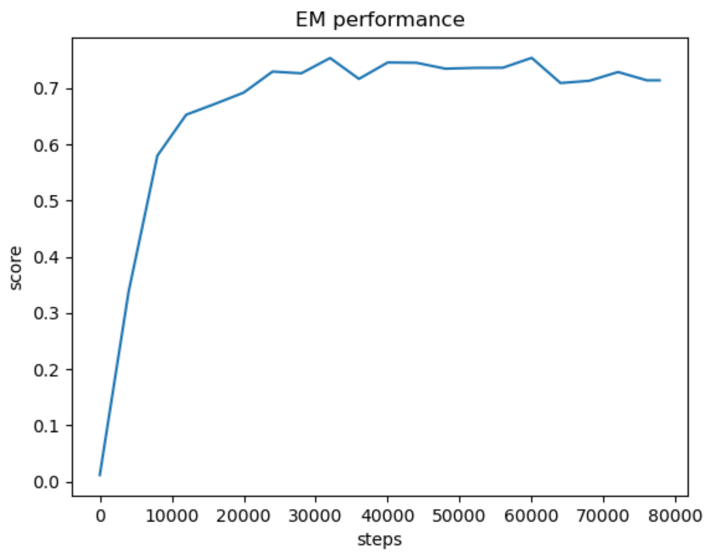
Q3: Curves

使用 matplotlib 套件繪出，每訓練 4000 步得到一組資料，且為第一個 epoch。

1. Plot



a. learning curve of EM



b. learning curve of F1