AI CUP 2023 春季賽

真相只有一個: 事實文字檢索與查核競賽報告

TEAM_3598
黃學智 (隊長)、李承哲、陳宥橋、朱祐麟

01.

題目資料介紹與資料分析

題目資料介紹與資料分析

在模型訓練中，我們總共擁有兩份訓練資料集。為了增加訓練資料的量，我們將兩份資料集合併成一份訓練資料集，以擁有更多樣本以加強模型學習並提高預測準確性。

合併後共有 11647 筆資料，每一筆由 “id”, “label”, “claim”, 和 "evidence" 四欄所組成。

如：

{
  "id": 2663,
  "label": "refutes",
  "claim": "天衛三軌道在天王星內部的磁層，以《 仲夏夜之夢 》作者緹坦妮雅命名。",
  "evidence": [[[4209, 4331, "天衛三", 2]]]
}

題目資料介紹與資料分析

各欄定義為：

- id: 索引號
- label: 三種 label 「 "supports", "refutes", 和 "NOT ENOUGH INFO" 」 代表了該筆資料是真是假；亦或是證據不足無法判定。
- claim: 該筆資料內容
- evidence: 用以判定 label 之證據，為從 wiki 頁面集中取出之句子集合。

題目資料介紹與資料分析

此外，我們擁有二十四份 wiki 頁面集。每一頁面皆由三欄 “id”, “text”, “line” 所組成，例如：

{

"id": "力學",

"text": "力學 （ mechanics ） 是物理學的一個分支 ， 主要研究能量和力以及它們與物體的平衡 、 變形或運動的關係 。",

"lines": "0\t力學 （ mechanics ） 是物理學的一個分支 ， 主要研究能量和力以及它們與物體的平衡 、 變形或運動的關係 。\n1\t"

}

因此，任務目標即為從頁面集內容判斷每筆資料的 label。

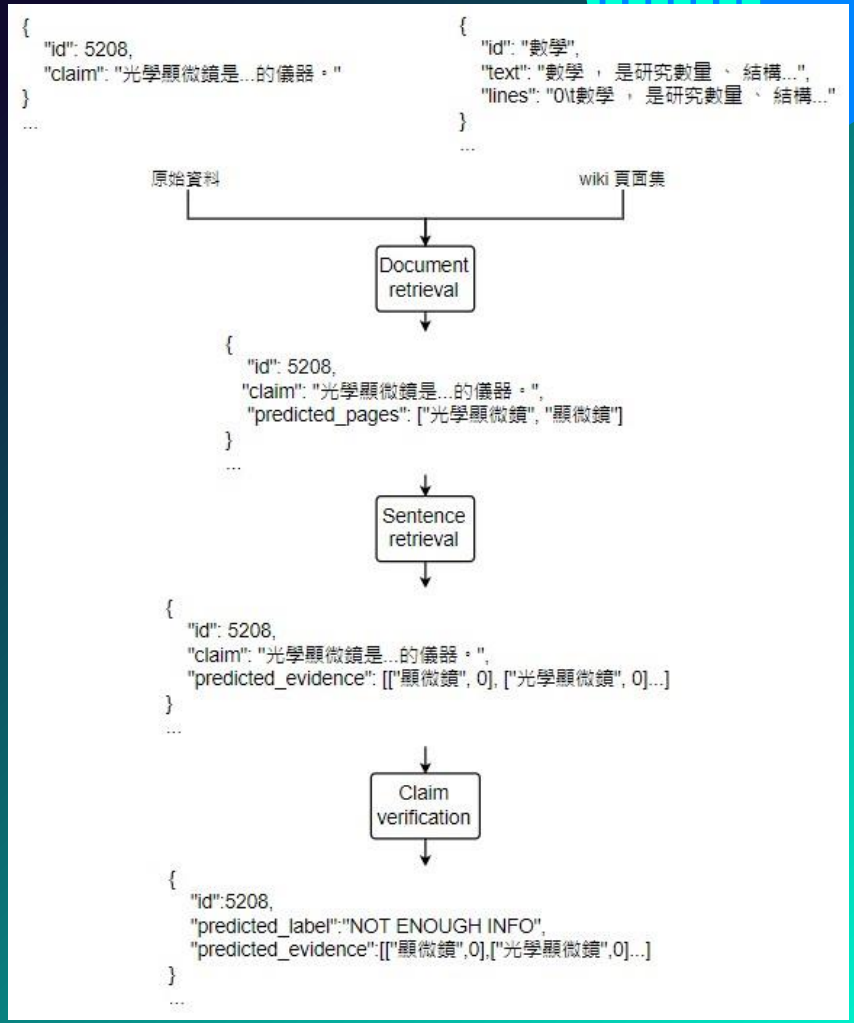
02.

嘗試方法的列出與比較與最佳方法

方法

首先，這次比賽的基本流程可分為三大部分。

1. Document retrieval
2. Sentence retrieval
3. Claim verification



方法

1. Document retrieval

   分析傳入的資料集與 wiki 頁面集，列出預測可能足以驗證資料真偽的頁面。

1. Sentence retrieval

   分析傳入的資料集與預測頁面集，列出預測可能足以驗證資料真偽的句子。

1. Claim verification

   分析傳入的資料集與預測句子集，列出預測資料真偽。

資料處理

在劃分訓練資料和驗證資料時，我們測試了兩種不同的比例，8:2 和 10:1。

我們發現比例設置為 10:1 時，模型在預測測試資料時表現更優秀，因此最後設為 10:1。

Document retrieval

此部分使用 tf-idf (Term Frequency-Inverse Document Frequency) 作為主要的文檔檢索演算法。

我們使用了 TfidfVectorizer，其主要超參數為：

- min_df = 1
- max_df = 0.8
- use_idf = True
- sublinear_tf = True
- ngram_range = (1,2)

```
Precision: 0.6024289658247152
Recall: 0.9000830232465085
```

Document retrieval

訓練流程：

- 我們先使用 jieba 進行分詞，取得所有 claim 與 wiki-pages 的 "text" 的 tf-idf
- 進行比較取出前 5 個內容相似的 wiki 條目做為 predict_page。

Sentence retrieval

在第二階段，我們使用了以下參數進行 model 的訓練：

- Training Batch: 32
- Val Batch: 256
- Optimizer: AdamW
- Scheduler: ReduceLROnPlateau (factor=0.1, patience=2, mode='min') *step with val loss
- Loss function: BCEWithLogitsLoss (with class weight)

Sentence retrieval

資料處理：

- 輸入為 Claim + WikiPageName + 上一句 + 本句 + 下一句
- Training data：
  - label 1
    - 在 evidence 中 page 存在的句子，就會標示為 label 1。
      - ~~Claim 1 Sent 1 + Sent 2 + Sent 3~~
      - Claim 1 Sent 1
      - Claim 1 Sent 2
      - Claim 1 Sent 3

Sentence retrieval

- label 0
  - 在 evidence 中 page 不存在的句子，就會標示為 label 0。
    - 例如：一個 page 有 5 個句子，sent 1 和 sent 2 存在，那 sent 3 到 5 就是 label 0。
    - 僅標至「0 與 1 的數量相同」
  - 接著在 predicted_pages 中取尚未標示為 label 0 的句子。為了避免 label 0 的資料過多，此時滿足條件的句子其中只有 10% 機率會被標示為 label 0 並納入。
- Val Data
  - 同上，但不會做計數，所有存在的句子標為 1，不存在的標為 0。即不存在的不會因為機率或配合 1 的數量做刪減。

Sentence retrieval

- 最後，訓練資料集大小為：
  - train_preprocessed length: 49374
    - 0 37929
    - 1 11445

Sentence retrieval

訓練流程：

- 將 Training Data 分成 Train 跟 Dev，然後把 Data 調整成我們的格式，轉成 pandas dataframe 然後包裝成我們的 Custom Dataset，在 Custom Dataset 中我們會把句子拼合增加 Special Token 然後輸入 Tokenizer 供給模型訓練。
- 把 Dataset 包裝成 DataLoader，訓練用的 batch size 為 32，然後會隨機排次序，檢測用的 batch size 為 256。
- 載入 hfl/chineses-lert-large，然後在 pooler_output 增加 Dropout（機率為 0.3），最後把輸出改為 Binary Classification。

Sentence retrieval

- 使用 AdamW 優化模型參數，然後使用 ReduceLROnPlateau 配合 Validation loss 設定 learning rate（一開始為 2e-5）。
- Loss function 我們使用 BCEWithLogitsLoss 及計算 class weight，然後每跑一個 batch 就會更新參數一次。
- 我們使用了 classification report 及自行計算平均 precision、recall、f1 score，我們會把最高平均 f1 score 的模型儲存並後面使用。

Claim verification

在第三階段，我們一樣採用了 LERT 作為我們的訓練模型。

主要超參數為：

- Batch Size: 16
- Seed: 42
- Learning Rate: 2e-5
- Max Sequence Length: 256

Claim verification

訓練流程：

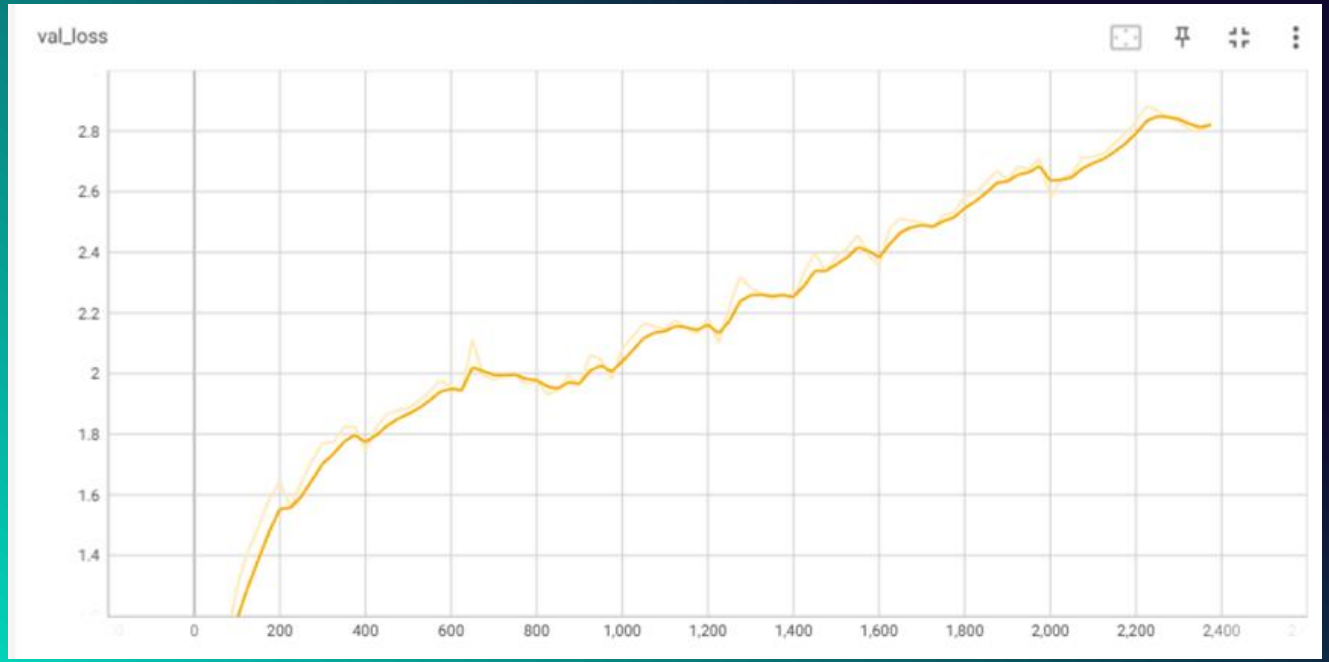
- 資料前處理：我們使用了 Hugging Face 庫中的 AutoTokenizer 從預訓練模型中載入分詞器（tokenizer）。並使用 AicupTopkEvidenceBERTDataset 類別來處理訓練資料和驗證資料，設定了最大序列長度（MAX_SEQ_LEN）。
- 資料載入：我們使用 DataLoader 來將訓練資料和驗證資料進行批次（batch）載入。設定了訓練資料的批次大小（TRAIN_BATCH_SIZE）和驗證資料的批次大小（TEST_BATCH_SIZE）。
- 我們使用 AutoModelForSequenceClassification 載入預訓練模型 hfl/chinese-lert-large。
- 我們一樣使用 AdamW 優化器來優化模型參數，並根據訓練步驟調整訓練的學習率（LR）。

Claim verification

- 我們使用迴圈進行多個訓練迭代。在每個訓練迭代中，我們將批次資料送入模型進行預測，並計算損失（loss）。然後，我們根據損失進行反向傳播（backpropagation）和參數更新。同時，我們記錄訓練損失和準確率到 TensorBoard 中。
- 在每個驗證步驟（VALIDATION_STEP），我們使用驗證資料集（eval_dataloader）對模型進行評估，並記錄評估結果到 TensorBoard 中。 同時，我們根據驗證準確率（val_acc）保存模型的 checkpoint。

Claim verification

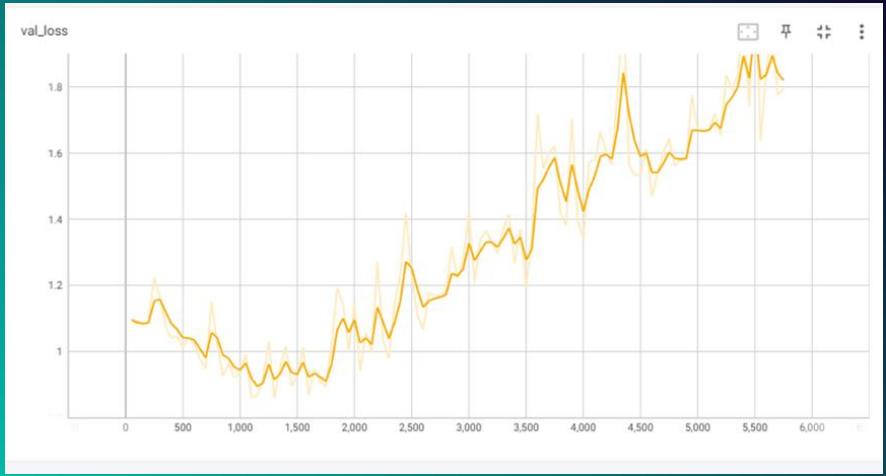
- 在此部分，我們發現在標籤為 "Not enough info" 的情況下，後面的證據欄位會是空的。這對我們的模型訓練會產生不良的影響，使得驗證損失（val_loss）不斷上升，如下圖：



Claim verification

方法一：

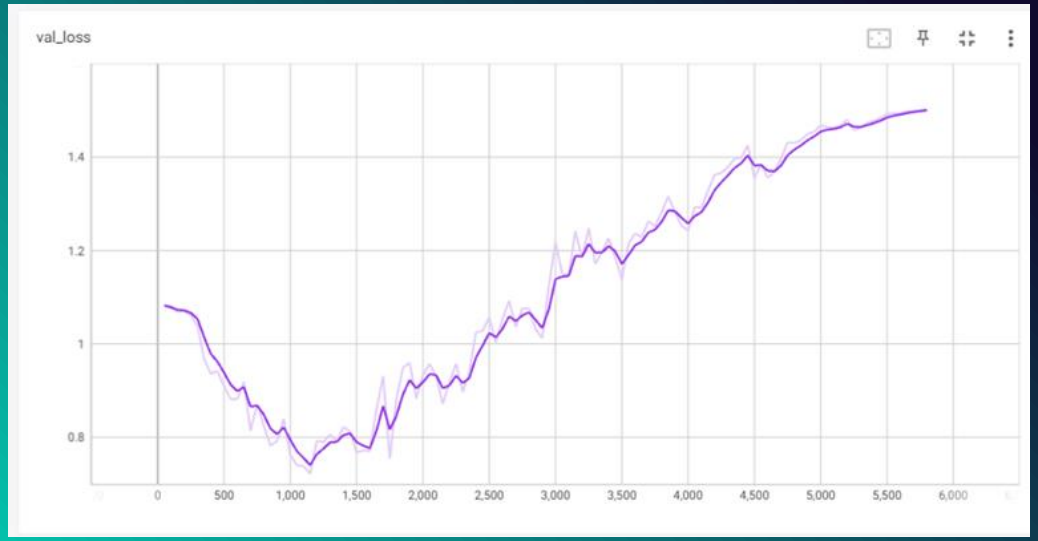
當我們的模型訓練遇到 "Not enough info" 標籤時，我們選擇使用 predicted_evidence 欄位中的第一個元素作為訓練時的證據列表（evidence_list）。為了避免 predicted_evidence 欄位也為空的情況，我們在前兩階段設置了至少要選出一個 predicted_evidence 的門檻。實作後，驗證損失（val_loss）的變化如下:



Claim verification

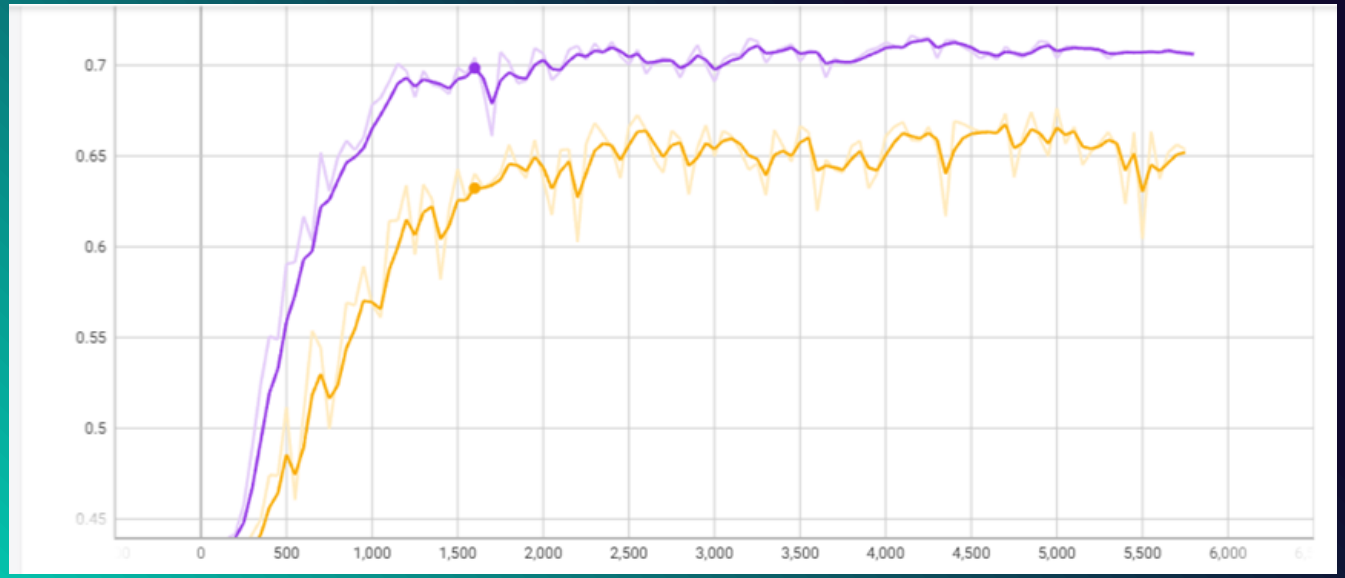
方法二：

我們直接取 predicted_evidence 作為訓練用的 evidence_list，儘管正確的 evidence 可能不在 predicted_evidence 中，但可以達到對三種結果（support、refute 和 not enough info）的公平性。測試結果如下圖:



Claim verification

兩方法比較 (accuracy)：



Claim verification

兩方法比較 (loss)：

最佳成績

- Public: 0.598
- Private: 0.689

03.

心得

心得

這次 FDA 與 AICUP 的經歷並不容易。我們在處理龐大的資料集和建立準確的模型時遇到了許多挑戰。除了自己的知識與能力的限制外，硬體運算資源的獲取也是一大挑戰。

無論如何，我們成功贏得了第五名的佳績，克服了萬難，也從半生不熟的資料處理入門者，變成了相信自身能力與經驗的參賽者。

我們團隊珍惜這次的經歷和成果。這不僅是一個技術的勝利，更是我們團隊合作和成長的證明。我們期待著未來的機會，在機器學習領域能繼續不斷學習和成長，以探索更多的知識。

最後，非常感謝教授與助教為我們帶來這學期 FDA 的課程，給我們這次寶貴的學習機會。

04.

分工表

分工表

- 李承哲：Document retrieval、Claim verification
- 黃學智：Sentence retrieval
- 朱祐麟：Mascot、PPT、Report

THANKS

Do you have any questions?