AI CUP 2021 醫病決策預判與問答成果報告書

1. 演算法說明

QA task:

圖 1　QA task 流程圖

1. Input：將訓練資料和測試資料讀入。
2. 文本斷句：先將訓練資料和測試資料每一筆的對話語境做斷句，以「人物：.... 」 為一句，切成多個句子。
3. 關鍵句萃取 ： 每篇文章經過斷句後，獲得多個句子，從答案選項分別以字串比對的方式，從多個句子裡挑選關鍵句，並使用停用詞將部分文字不做字串比對，例如:醫師、個管師...等等。每個答案各別取最相近的 2 個句子，3 個答案總共會取出 6 句，若有重複的句子則不取出。最後對話語境會被萃取成只有關鍵句的文本，當作模型最後輸入的對話語境。
4. 問句轉換 ： 把訓練資料和測試資料中的問句做篩選，部分問句的選項與標籤可能有全形的文字，一律轉為半形。若問句裡面含有「有誤」、「不是」...等反義詞，將其改為「正確」、「是」..... 等等，並把答案標籤改為其他選項，將改變後的問題及答案當作模型的輸入。
5. 模型訓練及預測 ： 使用的模型為 MacBERT [1]，將前處理過後的對話語境、問題及答案轉為簡體，將外部資料 C3 資料集 [2] 與訓練資料相疊，一起放入模型中訓練，如圖 2 所示。每段對話語境、問題(question)、答案(answer)使用[SEP]相接，最後經過 softmax 取得每個答案輸出的機率值。[3]

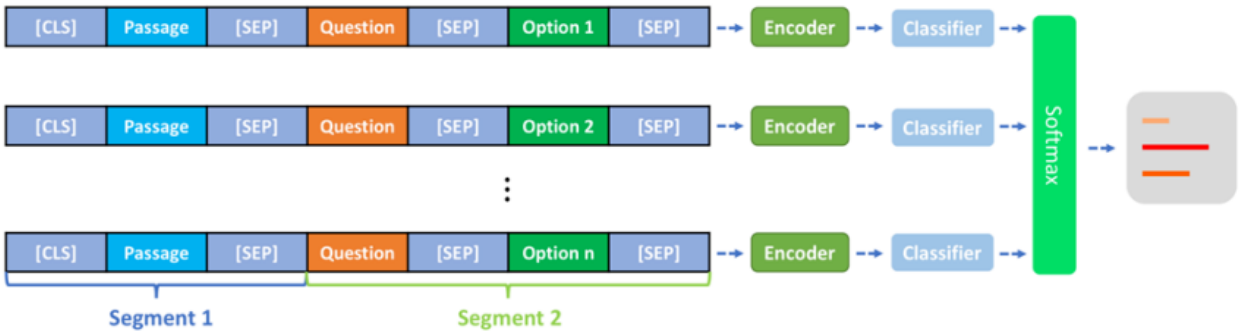

圖 2　QA task model 架構圖

6. 模型輸出後處理 ： 根據問題將三個選項的機率值做處理，每個問題選擇最高機率的選項(argmax)，作為最後輸出，若在前處理中問句中含有反義詞，則取最低機率的選項(argmin)為最後輸出。

7. Output：輸出 qa.csv 檔案。

Risk task:

圖 3　Risk task 流程圖

1. Input：將訓練資料和測試資料讀入。
2. 文本斷句：先將訓練資料和測試資料每一筆的對話語境做斷句，以「人物：.... 」 為一句，切成多個句子。
3. 關鍵句萃取：擷取醫生、個管師的對話部分，並將「人物：」去除。
4. 句數截斷並合併：將上述結果進行截斷。首先將所有句子字數低於 10 個字的句子刪除，並計算剩下句子的總字數，若前 2/3 句子總數大於 500 字，則以句子為單位，取中間前 2/3 句子總字數 500 字以內的句子當成資料，並將擷取的句子合併起來。
5. 模型訓練及預測：使用的模型為 RoBERTa-large [4]，如圖 4 所示，將整理好的資料隨機取 90%當成 training data，剩下 10%當成 evaluation data，輸入模型中訓練及預測，最後輸出兩個機率值。[5]

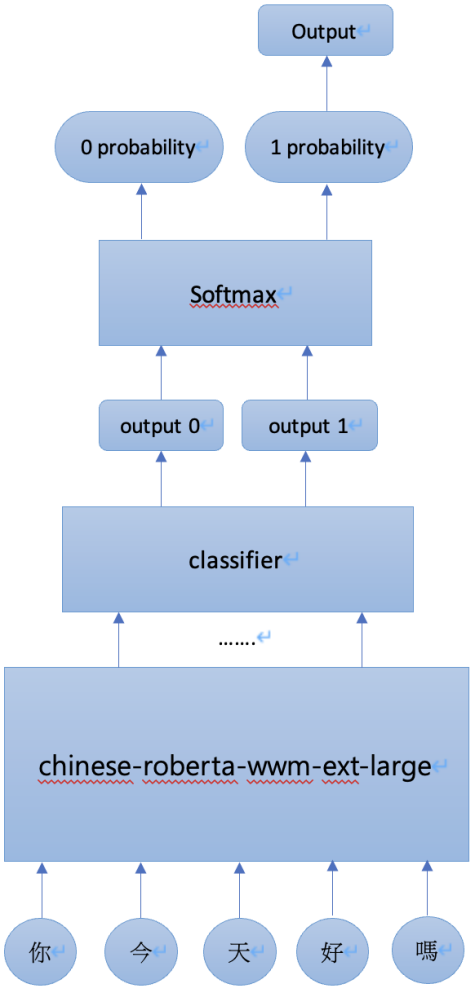

圖 4　Risk task model 架構圖

6. 模型輸出後處理：將模型輸出的兩個機率經過 Softmax，取 label 1 的輸出當預測值。
7. Output：輸出 decision.csv 檔案。

2. 工具說明

QA task:

- 程式語言版本：Python 3.7.0
- 作業系統：Windows 10、Linux
- pip install 套件：
  - Tensorflow-gpu 1.13.1、Transformers 4.6.0、Opencc 1.1.1
- conda install 套件：
  - pytorch 1.7.1、cudatoolkit 10.1

程式碼執行方法:

1. 先將訓練資料、外部資料、測試資料 Train_qa_ans.json 、c3 data set、Test_QA.json 放入 data 資料夾、執行 qa_process.ipynb，執行完會在 input_data 資料夾獲得 c3-d-train、c3-m-train、c3-d-dev、c3-m-dev、c3-d-test、c3-m-test 等 6 個 json 檔。
2. 將下載好的 model checkpoint 放入 input_data 資料夾、若要重新訓練則將預訓練模型下載至 chinese_pretrain_mrc_macbert_large 資料夾。
3. 執行 C3_finetune.py ，GPU 需自行設定張數，這裡選擇 gpu_ids 2,3，GPU 記憶體約 60000MB 才能執行，執行方法與參數設定如下：
   python C3_finetune.py --task_name c3 --gpu_ids 2,3 --do_eval --data_dir input_data --vocab_file chinese_pretrain_mrc_macbert_large/vocab.txt --bert_config_file chinese_pretrain_mrc_macbert_large/bert_config.json --init_checkpoint input_data/model_best.pt --max_seq_length 512 --train_batch_size 4 --eval_batch_size 16 --learning_rate 2e-5 --num_train_epochs 4.0 --output_dir input_data --gradient_accumulation_steps 1 若須重新訓練，則把--init_checkpoint 改為預訓練模型 chinese_pretrain_mrc_macbert_large/pytorch_model.bin，並加上--do_train，程式原始碼資料夾內已附有測試集模型輸出結果 logits_test.txt，可直接進行下一步後處理。
4. 訓練與測試完畢 input_data 資料夾會得到 logits_test.txt，執行 qa_output.ipynb，將模型輸出 logits_test.txt 轉為能上傳的 qa.csv 檔，並存在 output 資料夾裡。

♦ 訓練好的模型（chinese_pretrain_mrc_roberta_wwm_ext_large）下載連結：

https://drive.google.com/drive/folders/1n1Fve1WfUAQpb3IibBsVKpokmBQH5A6_?usp=sharing

- 預訓練模型（luhua/chinese_pretrain_mrc_roberta_wwm_ext_large）連結:
  https://huggingface.co/luhua/chinese_pretrain_mrc_roberta_wwm_ext_large、
  https://drive.google.com/drive/folders/1EvswU9SkMGOztxFbkUDTT4m2NvZ492Kt?usp=sharing

Risk task:

- 程式語言版本：Python 3.7.0
- 作業系統：Linux
- pip install 套件：
  - Transformers 4.6.1
- conda install 套件：
  - pytorch 1.6.0、torchvision 0.7.0、cudatoolkit 10.1.243、scikit-learn 0.23.2

程式碼執行方法：

1. 將訓練資料 Train_risk_classification_ans.csv 及測試資料 Test_risk_classification.csv 放入「aicup_risk_data」資料夾中。
2. 將下載的 model checkpoint 「risk_model.pt」放在「aicup_risk_model_state」資料夾中
3. 若需要修改參數，可修改之參數包括：
   I. 訓練：python aicup_risk_train.py --GPU 0 --MODEL_NAME hfl/chinese-roberta-wwm-ext-large --SAVE_STATE_NAME risk_model.pt --MODEL_SAVE_PATH aicup_risk_output --BATCH_SIZE 8 --EPOCHS 4 --LR 2e-5 --MAX_SEQ_LENGTH 512
   II. 測試：python aicup_risk_test.py --GPU 0 --MODEL_NAME hfl/chinese-roberta-wwm-ext-large --LOAD_STATE_NAME risk_model.pt -- BATCH_SIZE 8 --MAX_SEQ_LENGTH 512
4. 若要訓練，則執行「aicup_risk_train.py」，則在 aicup_risk_output 資料夾中輸出 risk_model.pt、risk_model_structure.txt。
5. 若要測試，則執行「aicup_risk_test.py」。則在 aicup_risk_output 輸出 decision.csv

- 訓練好的模型(chinese-roberta-wwm-ext-large)參數下載連結：
  https://drive.google.com/file/d/1InA3beOGjMbFiixz3T9hkdiXw7GfQzC2/view?usp=sharing

- 預訓練模型連結(hfl/chinese-roberta-wwm-ext-large):
  https://huggingface.co/hfl/chinese-roberta-wwm-ext-large
  https://github.com/ymcui/Chinese-BERT-wwm

3. 流程說明

QA task:

1. 文本斷句、關鍵句萃取、問句轉換 ： 執行 qa_process.ipynb，執行完獲得文本斷句、關鍵句萃取完成的訓練資料並做問句轉換，最後再與外部資料相疊 。
2. 模型訓練及預測 ： 執行 C3_finetune.py，輸出機率值檔案 logits_test.txt 。
3. 模型輸出後處理 ： 執行 qa_output.ipynb，將模型輸出檔 logits_test.txt 轉為能上傳的 qa.csv 檔，並存在 output 資料夾裡 。

Risk task:

1. 文本斷句、關鍵句萃取、問句轉換 ： 執行 aicup_risk_train.py 中的 get_data( )。
2. 模型、optimizer、scheduler 建構：執行 aicup_risk _train.py 中的 create_model( )、create_opt( )、create_lr_scheduler( )。
3. 模型訓練: 執行 aicup_risk _train.py 中的 train( )，則在 aicup_risk_output 資料夾中輸出 risk_model.pt、risk_model_structure.txt。
4. 模型測試 ： 執行 aicup_risk _test.py，則在 aicup_risk_output 輸出 decision.csv。

4. 組態說明 (e.g.環境設定、參數設定)

QA task	
環境設定	Python 3.7.0、Tensorflow-gpu 1.13.1、Transformers 4.6.0、Opencc 1.1.1、pytorch 1.7.1、cudatoolkit 10.1
參數設定	learning rate=2e-5、max_seq_length=512、num_train_epochs=4.0、train_batch_size=4、eval_batch_size=16、gradient_accumulation_steps=1、data_dir=input_data output_dir=input_data、init_checkpoint=input_data/model_best.pt、random seed=345
Risk task	
環境設定	Python 3.7.0、Transformers 4.6.0、pytorch 1.6.0、torchvision 0.7.0、cudatoolkit 10.1.243、scikit-learn 0.23.2
參數設定	GPU=0、MODEL_NAME=hfl/chinese-roberta-wwm-ext、LOAD_STATE_NAME=risk_model.pt、SAVE_STATE_NAME=risk_model.pt、MODEL_SAVE_PATH=aicup_risk_output、BATCH_SIZE=8、EPOCHS=4、LR=2e-5、

	MAX_SEQ_LENGTH=512、random seed 未設定（取用在開發集中成績最好模型）