

AI CUP 2021

醫病決策預判與問答成果報告書

1. 參賽隊伍資訊

隊名	NTUNLP_LionTea		
是否參加創意獎選拔 ■ Yes □ No			
	*姓名	*學校	*系/所
*指導教授	無指導教授		
*隊員	阮明皓 MING-HAO JUAN	國立臺灣大學 National Taiwan University	資訊工程學系暨研究所 Department of Computer Science and Information Engineering
	韓秉勳 BING-SHIUN HAN	國立臺灣大學 National Taiwan University	資訊工程學系暨研究所 Department of Computer Science and Information Engineering
	趙禹誠 YU-CHEN CHAO	國立臺灣大學 National Taiwan University	資訊管理學系暨研究所 Information Management

2. 演算法說明

a. 醫病問答

i. 流程簡述

- 訓練(train):
 - 1) 資料清理: 將所有字母字元轉半形以及統一標點符號
 - 2) 檢索預處理: 使用 BM25 資訊檢索演算法和中研院的開源套件 CkipTagger 的斷詞功能, 用問題及各選項從對話中檢索重要的篇章
 - 3) 模型訓練: 使用 Huggingface 的 BertForMultipleChoice 模型(MacBERT large)進行訓練, 訓練前會切出 10% 的資料作為驗證集
- 預測(test): 和訓練一樣先做資料清理及檢索預處理, 使用 BertForMultipleChoice 預測
- 集成(ensemble): 將在驗證集中答題正確率高於 0.65 的所有模型的預測進行投票

ii. 檢索預處理簡述

• BM25

一種用來計算搜索 Q 包含搜尋字 q_1, q_2, \dots, q_i 和文本 D 之間相關性的演算法, BM25 是一種詞袋檢索功能, 是 TF-IDF 資訊檢索演算法的衍生。數學式如下, 給定一個查詢 Q 包含搜尋字 q_1, q_2, \dots, q_i , 文檔 D 的 BM25 分數為:

$$\text{score}(D, Q) = \sum_{i=1}^n \text{IDF}(q_i) \cdot \frac{f(q_i, D) \cdot (k_1 + 1)}{f(q_i, D) + k_1 \cdot \left(1 - b + b \cdot \frac{|D|}{\text{avgdl}}\right)}$$

其中 $f(q_i, D)$ 表示 q_i 出現在文檔 D 中的頻率

$|D|$ 是文檔 D 的長度(以字為單位)

avgdl 是文本集合中文檔的平均文檔長度

k_1 和 b 是自由參數

$\text{IDF}(q_i)$ 是查詢詞 q_i 的 IDF 權重, 它通常計算為:

$$\text{IDF}(q_i) = \ln\left(\frac{N - n(q_i) + 0.5}{n(q_i) + 0.5} + 1\right)$$

其中 N 是文本集合的文本總數

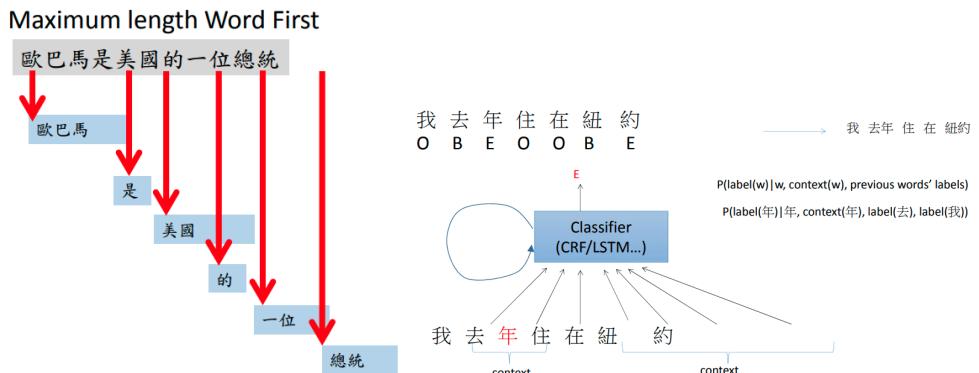
$n(q_i)$ 是包含 q_i 的文檔數

• 中研院開源套件 CkipTagger

- 模型: CkipTagger 開源中文處理工具包含(WS)斷詞、(POS)詞性標注、(NER)實體辨識功能, 在斷詞與詞性標記的表現大幅超越結巴系統。CkipTagger 使用 Cross-BiLSTM-CNN 模型進行訓練, 主要分別捕捉 low-level 及 high-level 隱藏特徵(hidden features), 使得得以解決 BiLSTM 的 XOR 限制。數學公式如下: 將輸入序列 X 分別過正向(forward)跟反向(backward)的 LSTM, 之後將兩者的隱藏狀態(hidden states)進行行串連(row-wise concatenation), 接續將串連的隱藏狀態分別過正向跟反向的 LSTM, 最後將結果再次進行串連。

$$\begin{aligned}\vec{H}^1 &= \overrightarrow{LSTM}_1(X) \\ \overleftarrow{H}^3 &= \overleftarrow{LSTM}_3(X) \\ \vec{H}^2 &= \overrightarrow{LSTM}_2(\vec{H}^1 || \overleftarrow{H}^3) \\ \overleftarrow{H}^4 &= \overleftarrow{LSTM}_4(\vec{H}^1 || \overleftarrow{H}^3) \\ H &= \vec{H}^2 || \overleftarrow{H}^4\end{aligned}$$

- 斷詞技巧：同時使用單詞等級（Word-level）及字符等級（Character-level）方法，在單詞等級上使用長詞優先（Maximum length）動態規劃查找最大概率路徑；字符等級則是使用序列標註（Character Sequence Labeling）。



- 訓練資料：使用中央社、維基（用舊版套件先進行斷詞）及 ASBC（為期近 10 年人工標記）為文本進行詞向量（word embedding）計算。

iii. 模型簡述

- 使用 Huggingface 的 BertForMultipleChoice (hfl/chinese-macbert-large)，此套件會依據輸入的模型名字自動帶入使用的參數及設定，本組在此使用 MacBERT large 的預訓練模型。此套件模型架構為預訓練模型加上 1 DropOut 層及 1 全接層。
- MacBERT (MLM as correction BERT)
 - 使用 pretrained model : hfl/chinese-macbert-large (<https://huggingface.co/hfl/chinese-macbert-large>)
 - 模型參數及架構

	BERT	BERT-wwm	RoBERTa-wwm	ELECTRA	MacBERT
Word #	0.4B	5.4B	5.4B	5.4B	5.4B
Vocab #	21,128	21,128	21,128	21,128	21,128
Hidden Activation	GeLU	GeLU	GeLU	GeLU	GeLU
Optimizer	AdamW	LAMB	AdamW	AdamW	LAMB
Training Steps	?	2M	1M	2M	1M
Init Checkpoint	random	BERT	BERT	random	BERT

Table 3: Training details of Chinese pre-trained language models.

MacBERT 由哈工大和科大訊飛提出，用谷歌官方的 Chinese BERT-base 進行參數初始化，同樣使用 BERT 架構，並用相同的預訓練方式，唯一在 MLM (Masked Language Model) 訓練上做了一些調整，如下：

- 採用基於分詞的 n-gram 掩碼（masking），1-gram~4gram 掩碼的概率分別是 40%、30%、20%、10%

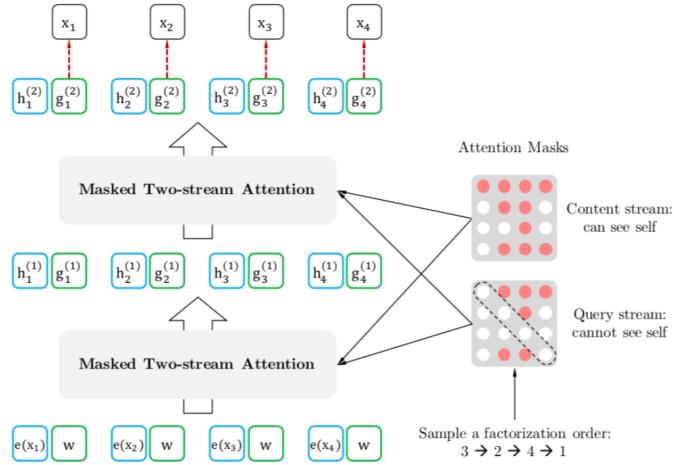
	Chinese	English
Original Sentence + CWS + BERT Tokenizer	使用语言模型来预测下一个词的概率。 使用语言模型来预测下一个词的概率。 使用语言模型来预测下一个词的概率。	we use a language model to predict the probability of the next word. - we use a language model to pre ##di ##ct the pro ##ba ##ility of the next word .
Original Masking + WWM ++ N-gram Masking +++ Mac Masking	使用语言[M]型来[M]测下一个词的概率。 使用语言[M][M]来[M][M]下一个词的概率。 使用[M][M][M]来[M][M]下一个词的概率。 使用语法建模来预见下一个词的几率。	we use a language [M] to [M] ##di ##ct the pro [M] ##ility of the next word . we use a language [M] to [M][M] the [M][M][M] of the next word . we use a [M][M] to [M][M][M] the [M][M][M][M] next word . we use a text system to ca ##lc ##ulate the po ##si ##bility of the next word .

Figure 1: Examples of different masking strategies.

- 採用同義詞取代 [MASK] 進行掩碼，同義詞基於 word2vec 相似度計算來獲取。在極少數情況下，當沒有相似的詞時，會使用隨機詞替換
- 掩蔽 15% 輸入單詞，其中的 80% 替換為同義詞，10% 將替換為隨機單詞，剩下的 10% 將保留原始單詞
- NSP 任務採用同 ALBERT 一樣的 序列順序預測 (sentence-order prediction, SOP) 任務，預測這兩個句子對是正序還是逆序
- 模型效果
 - 採用了三種類型的任務來進行評測，包括機器閱讀理解 (CMRC 2018、DRCD、CJRC) 、單句文本分類(ChnSentiCorp、THUCNews)和文本對分類(XNLI data、LCQMC、BQ Corpus)，在絕大多數的人任務上都達到 SOTA 效果 (略數據，詳見原論文) 。
- 下游任務：多選閱讀理解
 - 因為多選閱讀理解本身即是多元分類問題，故本組可直接使用 MacBERT 了解語意並利用全接層來輸出多元分類的機率。

b. 決策預判與風險評估

- i. 流程簡述
 - 訓練 (train) :
 - 1) 資料清理: 無
 - 2) 模型訓練: 使用 Huggingface 的 XLNetForSequenceClassification 模型 (XLNet base) 進行訓練，訓練前會切出 10% 的資料作為驗證集
 - 預測 (test) : 和訓練一樣先做資料清理及檢索預處理，並使用 XLNetForSequenceClassification 做預測
 - 集成 (ensemble) : 將在驗證集中 AUROC 高於 0.89 的所有模型的預測機率取平均
- ii. 模型簡述
 - 使用 Huggingface 的 XLNetForSequenceClassification (hfl/chinese-xlnet-base) ，此套件會依據輸入的模型名字自動帶入使用的參數及設定，本組在此使用 XLNet base 的預訓練模型。此套件模型架構為預訓練模型加上 1 全接層。
 - XLNet
 - 使用預訓練模型: chinese-xlnet-base (<https://huggingface.co/hfl/chinese-xlnet-base>)
 - 12層，768 隱藏狀態 (hidden state)，117M 參數量。
 - 此架構改善了 BERT 的一些缺點，例如：
 - BERT 的掩碼符號 [MASK] 並非原本句子中的符號，所以在下游任務訓練時與預訓練模型產生資訊不對等。
 - 無法確認同句子多個掩碼的交互關係
 - 句子中若有多個 [MASK]，無法確認兩者之間的關係。例如 bank crisis 兩個字若都被掩碼，BERT 無法確定兩者的交互關係。
 - XLNet 使用了 Two-Stream 自我注意力 (Self-Attention) 來避免此問題，利用內容流 (Content Stream) 學習上下文，查詢流 (Query stream) 則是代替 [MASK] 進行掩碼，欲了解實作細節請詳見原論文。



- AR (Auto regression) model 如 GPT 無法同時查看上下文關係，只能分別查看正向與反向序列。XLNet 使用了 Permutation Language Modeling (PLM) 的作法，將原本句子順序重新排列組合以得到所有字的前後文關係，因為採用內容流以及查詢流的方法，故能夠完成這樣的操作。

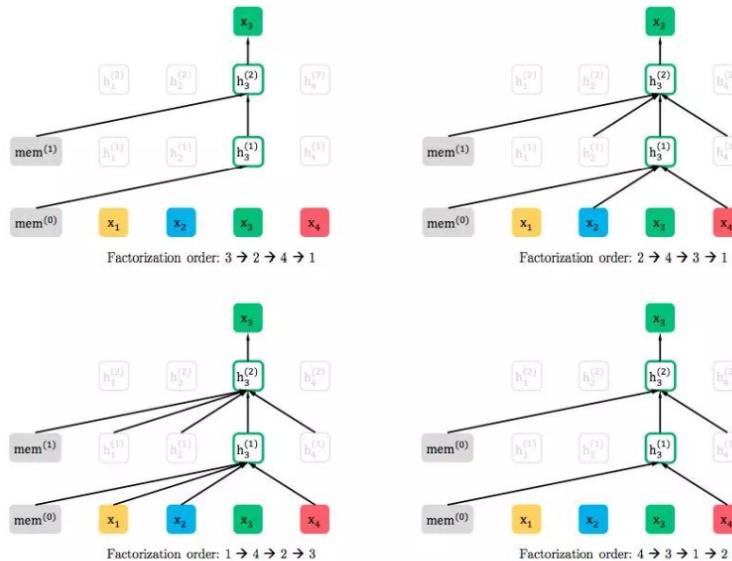


Figure 1: Illustration of the permutation language modeling objective for predicting x_3 given the same input sequence \mathbf{x} but with different factorization orders.

■ BERT 無法輸入長度大於 512 的序列

- 此競賽的閱讀文本長度平均落在 3000 左右，使用 BERT 無法將全部內容輸入模型，造成閱讀上的斷層。XLNet 利用 TransformerXL 的想法讓不同的文本片段能夠互相注意 (attention)，以達到讀取下一段文本資訊的效果。XLNet-base 能夠一次讀取 4096 長度的文本進入 model，能夠有更大機會讀懂一個長文並做出判斷。

● 下游任務：風險評估

- 因為判斷高低風險本身即是二元分類問題，故本組可直接使用 XLNet 了解語意並利用全接層來輸出二元分類的機率。

3. 工具說明

a. 程式語言版本

Python 3.8

b. 執行的作業系統

Ubuntu 18.04 (kernel: 4.15.0-65-generic)

c. 使用的每個套件的版本

本組使用 pipenv 的 Pipfile 進行套件管理，詳細套件版本如下：

torch==1.8.1

transformers==4.6.1

pandas==1.2.4

tqdm==4.61.0

scikit-learn==0.24.2

ckiptagger==0.2.1

tensorflow-cpu==2.4.1

wandb==0.10.32

d. 套件安裝方法

建議使用 pipenv install 安裝 Pipfile 中的套件，如果需要 log training metrics，可以使用 pipenv install --dev，會多安裝 wandb。pipenv install 後會建立一個虛擬環境，使用 pipenv shell 進入該環境便可以開始執行本組的程式。

e. 程式碼執行方法

以下僅列出可以重現本組測試集結果的步驟，如果要重新訓練，請參考程式碼隨附的 README 檔，以及下方組態說明中的參數設定。README 中也有以下的步驟，也可以直接參考 README 檔。

1) 按照 d. 套件安裝方法 建立執行環境。

2) 下載與解壓縮訓練好的模型：

bash download_model.sh

3) 將決策預判與風險評估(rc)及醫病問答(qa)的測試資料分別放在 data/rc/test.csv 與 data/qa/test.json。

4) 醫病問答的資料需要經過預處理：

```
python query_qa.py \
    --data_path data/qa/test.json \
    --model_name model_test.pkl \
    --processed_data_path data/qa/processed_test.json
```

5) 我們有集成(ensemble)3 個決策預判與風險評估的模型，醫病問答則是有 4 個，要一次跑所有模型，可以執行：

bash predict_rc.sh

bash predict_qa.sh

可以在 script 中調整 --batch_size 以符合 GPU 的記憶體大小

6) 最後，決策預判與風險評估會以平均做集成，醫病問答則是以投票做集成：

```
python ensemble.py \
    --task rc \
    --data_dir prediction/ \
    --pred_path prediction/rc_final.csv
python ensemble.py \
    --task qa \
    --data_dir prediction/ \
```

--pred_path prediction/qa_final.csv

這邊要特別注意的是，在醫病問答的集成投票這邊，因為在同票時本組是採用隨機選擇，而我們在比賽時忘記設隨機種子，所以無法完全重現比賽時的結果，但我們有信心在分數上會非常接近。另外，決策預判與風險評估的部分，可能會因為浮點數的精度，而有些許小誤差，不過在分數上應該也會非常接近。

7) 最後的預測結果會分別存在 prediction/rc_final.csv 及 prediction/qa_final.csv。

f. 訓練好的模型下載連結

建議使用 e. 2) 中的 download_model.sh 下載與解壓縮模型，這個 script 會一併處理路徑問題。若要直接下載模型 zip 檔，

rc_model.zip 網址：

https://drive.google.com/file/d/1GHO_PUPwRSYaHLgmWb8wKUD8dvsYw50w/view?usp=sharing

qa_model.zip 網址：

https://drive.google.com/file/d/1OH2J6m9j_sUmebpsW3aYrUDCgxJ-W-P/view?usp=sharing

請下載完解壓縮後放在 ckpt/ 下。

4. 流程說明

a. 醫病問答

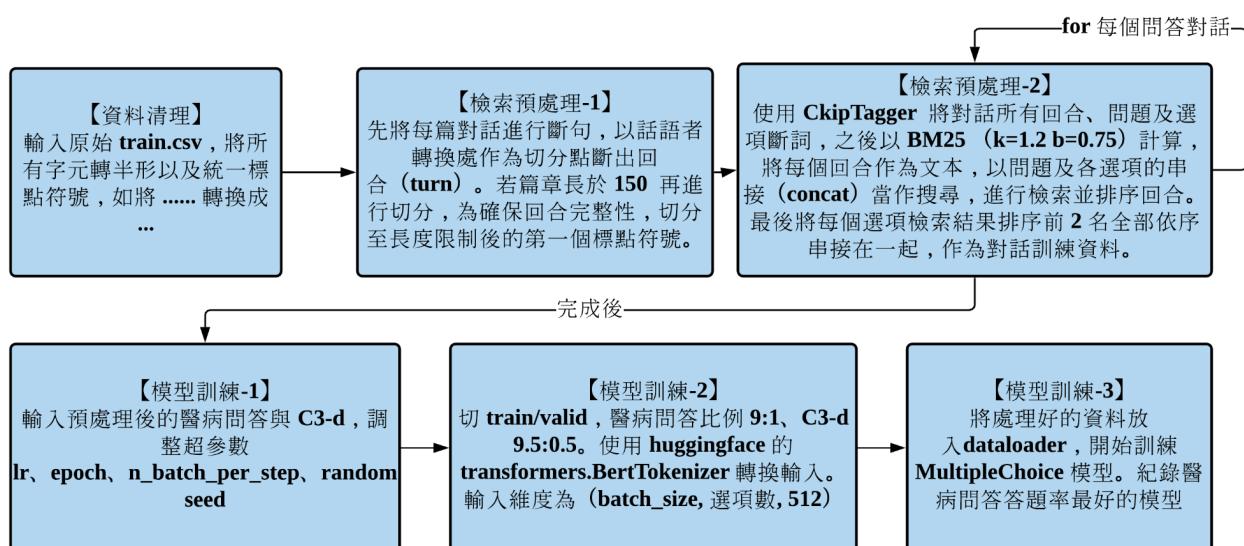
i. 資料集簡述

訓練加入其他中文閱讀理解多選資料集 C3 (<https://dataset.org/c3>)，其中只採用對話式 C3-d 的資料集，以下為資料簡述：

資料集	語言	型態	文本數量	問題數量	選項數量	文本平均長度
AICUP 醫病問答	繁中	對話式	346	695	3	2110.8
C3-d	簡中	對話式	8,140	9,571	3 或 4*	76.3

* 若選項數量為 4，則隨機移除一非答案的選項，以匹配 AICUP 資料集

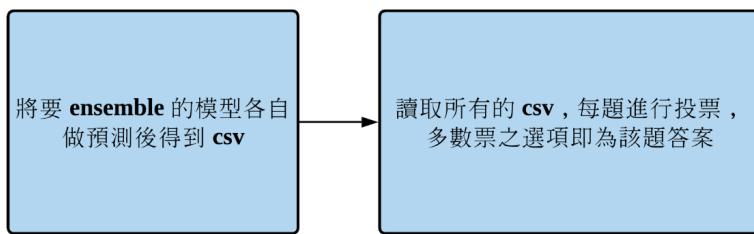
ii. 訓練流程



iii. 測試流程

同訓練流程，但無 C3-d 資料。

iv. 集成流程



v. 討論

- 為何決定使用 BM25 作為解決對話過長的問題？
 - 以下列舉當初嘗試過的解決方法以及最後結果

方法	結果
使用 longformer 無輸入長度限制	CUDA 記憶體不足，無法訓練
使用 xlnet 輸入長度限制 2048，直接截斷後半部/前半部多餘對話	訓練效果不好，最好答題率只有 0.4-0.5
使用 xlnet 輸入長度限制 1024，將對話切段，各別與問題選項輸入模型，最後再平均輸出 logits	訓練效果不好，最好答題率只有 0.4-0.5
使用 xlnet 輸入 1024 長度，先將對話進行 BM25 檢索預處理，在將對話問題選項輸入模型	雖然答題率依然沒有顯著提升，但人工檢查檢索出的回合大多都有覆蓋問題選項

- 為何決定使用 C3-d 作為額外訓練資料集？
 - 由於訓練資料只有近 700 多筆，相較與一般閱讀測驗資料集差了 1 個數量級多，因此本組推測，資料過少導致模型很快在訓練集上過度擬合，儘管使用 BM25 預處理後的對話高覆蓋問題選項，依然無法成功訓練。所以我們決定採用額外資料進行訓練，我們當初只選擇多選對話式閱讀測驗資料集，人選有簡中資料集 C3-d、英文資料集 DREAM，我們分別測試各別加入及全部加入，最後是只加入 C3-d 資料集的模型效果最好。

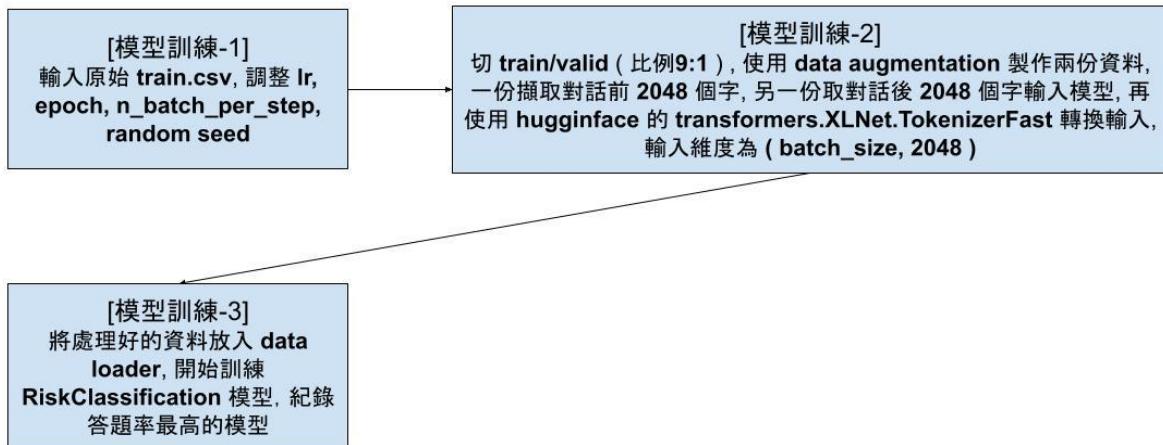
b. 決策預判與風險評估

i. 資料集簡述

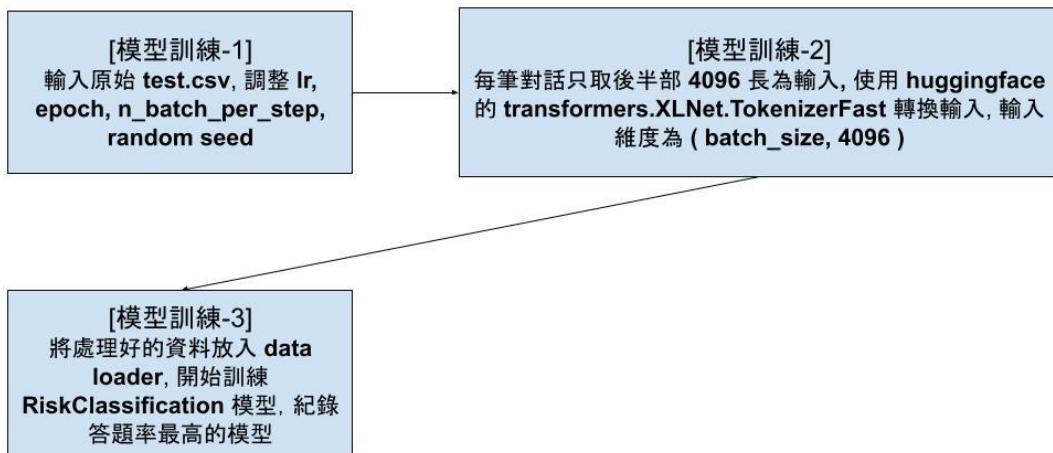
只取 AICUP 資料集訓練

資料集	語言	型態	文本數量	問題數量	文本平均長度
AICUP 風險評估	繁中	對話式	346	346	2110.8

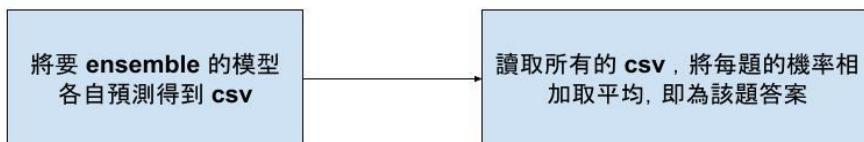
ii. 訓練流程



iii. 測試流程



iv. 集成流程



v. 討論

- 為何不做預處理？
 - 本組有嘗試使用醫病問答中的檢索預處理, 但最後結果仍不比原始資料加上 data augmentation 訓練效果來得好。
- 為何訓練時只取前 2048 或後 2048 個字, 而測試卻是取最後面 4096 個字?
 - 由於訓練時會使用較多記憶體, 取 2048 個字已達本組機器記憶體限制。然而若只取最後面 2048 個字沒有完整發揮訓練資料, 所以本組也將最前面 2048 個字加入訓練資料, 訓練準確率也確實提升。
 - 在測試時本組取 4096 個字是希望在可接受的預測速度與記憶體用量下取最多的內容, 由於對話大多不超過 4000 個字, 模型輸入已能覆蓋大部份的對話內容。

5. 組態說明

- a. 環境設定 - 請見 3. 工具說明
- b. 參數設定

本組醫病問答任務使用 4 個模型，決策預判與風險評估任務則是 3 個模型進行集成，各自參數列於下表：

集成執行請見 3. 工具說明

QA models					
Name	lr	n_batch_per_step	n_epoch	random seed*	valid_acc
qa_1	3.00E-05	16	20	0	0.7
qa_2	3.00E-05	24	5	0	0.6571
qa_3	3.00E-05	32	10	0	0.7571
qa_22	3.00E-05	16	10	1	0.6857

Risk classification models					
Name	lr	n_batch_per_step	n_epoch	random seed*	valid_auroc
rc_1	1.00E-05	16	30	0	0.9231
rc_4	5.00E-05	16	30	0	0.8951
rc_9	1.00E-05	16	30	4	0.9161

* random seed 改法：修改 utils.py 中 handle_reproducibility 的 torch.manual_seed

```
def handle_reproducibility(is_reproducible: bool = True) -> None:
    if is_reproducible:
        torch.manual_seed(4)
        torch.backends.cudnn.deterministic = True
        torch.backends.cudnn.benchmark = False
```

6. 外部資源與參考文獻

1. 論文

- Kai Sun et al., 2019. Investigating Prior Knowledge for Challenging Chinese Machine Reading Comprehension. arXiv preprint arXiv:1904.09679.
- YANG Zhilin et al. 2019. XLNet: Generalized Autoregressive Pretraining for Language Understanding. arXiv preprint arXiv:1906.08237.
- Peng-Hsuan Li et al. 2019. Why Attention? Analyze BiLSTM Deficiency and Its Remedies in the Case of NER. arXiv:1908.11046
- Yiming Cui et al. 2020. Revisiting Pre-trained Models for Chinese Natural Language Processing. arXiv:2004.13922

2. 網路資料

- Wikipedia. 2021. Okapi BM25. Wikipedia. https://en.wikipedia.org/wiki/Okapi_BM25
- 馬偉雲. 2019. 結合斷詞、詞性標記、實體辨識的一站式中文處理開源套件 - CkipTagger. https://linguistics.ntu.edu.tw/static/media/Ma_CkipTagger_1129.2f82ba88.pdf
- Kai Sun et al. 2019. Multiple-Choice Chinese Machine Reading Comprehension Dataset. <https://dataset.org/c3/>
- Peng-Hsuan Li and Wei-Yun Ma. 2019. CkipTagger GitHub Repo. <https://github.com/ckiplab/ckiptagger>
- MacBERT Model. <https://github.com/ymcui/MacBERT>
- WenWei Kang. 2019. 2019-NLP 最強模型 : XLNet. <https://medium.com/ai-academy-taiwan/2019-nlp%E6%9C%80%E5%BC%B7%E6%A8%A1%E5%9E%8B-xlnet-ac728b400de3>
- 忆臻.2020.什么是XLNet, 它为什么比BERT效果好 ? .<https://zhuanlan.zhihu.com/p/107350079>

7. 檢附隊員學生證明

1 阮明皓

國立臺灣大學 109 學年度第2學期在學證明

National Taiwan University

Certification of Enrollment

Spring Semester 2021

2021/6/25

學號 Student ID Number	R09922083
系所組 Department	資訊工程學系 Department of Computer Science and Information Engineering
姓名 Name	阮明皓 Ming-Hao Juan
年級 Year	1
學制 Program	碩士班 Master's Program
出生年月日 Date of Birth	87 年 1 月 18 日 January 18, 1998



國立臺灣大學 109 學年度第2學期在學證明

National Taiwan University

Certification of Enrollment

Spring Semester 2021

2021/6/25

學號 Student ID Number	R09922102
系所組 Department	資訊工程學系 Department of Computer Science and Information Engineering
姓名 Name	韓秉勳 Bing-Shiun Han
年級 Year	1
學制 Program	碩士班 Master's Program
出生年月日 Date of Birth	85 年 12 月 22 日 December 22, 1996



3 趙禹誠

