**AI-CUP 2019**

**人工智慧論文機器閱讀競賽之論文分類**

隊伍: SDML\_BetrayerLaBetrayer

成員: 陳義榮, 蔡秉辰, 黃秉迦

1. 環境

* 作業系統: Arch Linux
* 語言: Python 3.7.4
* 預訓練模型: [BERT-Base, Uncased](https://storage.googleapis.com/bert_models/2018_10_18/uncased_L-12_H-768_A-12.zip)
* 額外資料集: 無
* 使用的套件：
  + tensorflow == 2.0.0
  + keras-bert == 0.78.0
  + Keras == 2.2.4
  + pandas
  + nltk
  + tqdm

1. 資料處理

表現最好的模型僅採用 abstract 的資料和其 label。

但在實驗的過程，我們有試過使用 abstract、title、classifier，但效果並沒有比較好。

1. 模型架構

採用最經典的 BERT-based 的架構，最後在第一個輸出那邊接一個 fully connected layer 使其有三維的輸出。而判斷 Other 項的輸出為何，則是看前三維的輸出做決定。

1. 訓練方式

我們會先做 further pretrain，用來作為 BERT 的初始權重。然後先只訓練最後一層，再對整個模型做 fine-tune。Further pretrain 和 Fine-tune 的細節如下：

* Further pretrain:

先載入 google 官方訓練好的權重。接著用 training dataset 和 public testing dataset 來做 unsupervised training，降低 dataset 的 NSP, MLM loss。詳細參數如下：

* Fine-tune details

for i in {0, 1, 2}:

batch size = 12 + 4i

fine-tune last 3 - i encoders of BERT with 2 \* (i + 1) epochs

1. 分析&結論

由於fine-tune整個model很容易overfitting，所以我們只fine-tune後⾯幾個encoder。在那之前，避免classifier過差導致further pretrain model被backpropogate到更糟的地方，我們先對 classifier做了簡單的training。fine-tune多層能快速到達optimal solution但容易overfitting，所以 我們決定透過loop的⽅式fine-tune，使fine-tune的參數越來越少。fine-tune完後，再針對 classifier進⾏training就完成了。

1. 程式碼

Download the zip file from <https://drive.google.com/open?id=1PG2T8jkuWJH4eF3ZotqKDkcLj06So5CG>, and then extract it.

* Prepare data

1. mkdir data
2. cd data
3. Put "task2\_public\_testset.csv" and "task2\_trainset.csv" into data
4. wget <https://storage.googleapis.com/bert_models/2018_10_18/uncased_L-12_H-768_A-12.zip>
5. unzip -j uncased\_L-12\_H-768\_A-12.zip
6. You can also download our pretrained weight by

wget <https://www.csie.ntu.edu.tw/~b06902024/pretrain_bert_24_v2.weight>

* Start training

Run the following scripts:

cd src

python3 pretrain.py [pretrain\_weight\_path]

python3 train.py [pretrain\_weight\_path] [train\_weight\_path]

python3 submit.py [result.csv]

1. 使用的外部資源與參考文獻

* Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.
* CyberZHG. (2019, October 29). CyberZHG/keras-bert. Retrieved January 15, 2020, from https://github.com/CyberZHG/keras-bert.